Effectiveness of Shock Capturing Methods in the Discontinuous Galerkin/Flux Reconstruction Scheme for Computational Fluid Dynamics

Xavier Lesperance

A thesis

in

The Department

of

Mechanical, Industrial, and Aerospace Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Masters of Applied Science (Mechanical Engineering)

Concordia University

Montreal, Quebec, Canada

August 2021

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Xavier Lesperance**

Entitled: **Effectiveness of Shock Capturing Methods in the Discontinuous Galerkin/Flux Reconstruction Scheme for Computational Fluid Dynamics**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Mechanical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Hoi Dick Ng

_____ External Examiner
Dr. Rolf Wuthrich

_____ Examiner
Dr. Brian Vermeire

_____ Supervisor
Dr. Charles Basenga Kiyanda

Approved by _____
Dr. Sivakumar Naranswamy, MASc Program Director
Mechanical, Industrial and Aerospace Engineering

_____ 2022
Date

_____
Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Effectiveness of Shock Capturing Methods in the Discontinuous Galerkin/Flux Reconstruction Scheme for Computational Fluid Dynamics

Xavier Lesperance

The development of various numerical methods capable of accurately simulating fluid flow has evolved greatly over time. In the past years, discontinuous Galerkin methods have seen great interest for problems such as Large Eddy Simulations, Aeroacoustics, incompressible and even compressible flows. These methods attractiveness some from their ability to easily increase the order of accuracy thus yielding more precise solutions. These methods use higher order polynomials, which can easily be increased or decreased within the element, while allowing for discontinuities between elements. When shocks and discontinuities are present in a simulation, particular attention must be taken to avoid Gibbs phenomenon within the elements. This phenomenon occurs when steep gradients in the solution are present causing the solution to have erratic oscillations typically associated with the higher order terms of the integrating polynomial. These oscillations in turn lead to non-physical solutions such as negative pressures and therefore need to be controlled. A variety of methods have been developed to mitigate the oscillatory behavior of discontinuous Galerkin methods when steep gradients are present, a very promising method is the addition of artificial viscosity in order to diminish the effects of the non-physical oscillations. Adding a viscous term to the conservation equations being solved can inevitably lead to inaccurate solutions if it is added in excessive amounts. The balance between damping of the non-physical oscillations and minimizing the amount of artificial viscosity added can if the location of the shocks in the flow field is known. This intricate balance is achieved by ensuring that the functions used to find the areas of concern are not overlapping shock regions with smooth regions and when viscosity is added it is important that it is limited to ensure that it will not completely dissipate the real solution.

Acknowledgments

First and foremost, I would like to thank my supervisor Professor Charles Kiyanda for guiding me through my journey of exploring the world of detonations and computational fluid dynamics. I must also take this opportunity to thank Dr Brian Vermeire who assisted me throughout this entire process, without his invaluable insight and assistance none of this would have been possible. It is also important that I take the time to thank Chian Yan for assisting me with simulations using the Finite Volume solver. Secondly, I'd also like to take this time to thank my family for believing in me and always being there for me no matter what. This has been an incredible journey and I look forward to putting the skills I have learned during this degree to use in my career as an engineer.

Contents

List of Figures

List of Tables

Acronyms

aoa	Angle of Attack
CFD	Computational Fluid Dynamics
CFL	Courant-Fridrichs-Lewy
DOF	Degrees of Freedom
DG	Discontinuous Galerkin
FR	Flux Reconstruction
FV	Finite Volume
SV	Spectral Volume

Alphabetical Symbols

c	Wave Speed
C	Speed of Sound
C_p	Specific Heat for Constant Pressure
C_v	Specific Heat for Constant Volume
D	Diameter
E	Total Energy
e	Internal Energy
$\boldsymbol{F}$	Flux vector
$\hat{F}$	Numerical Flux
$\boldsymbol{f}_e$	Body Forces
H	Total Enthalpy
h	Specific Enthalpy
h	Element Length

$\overline{\overline{\boldsymbol{I}}}$ Unit Tensor

k Thermal Conductivity

M Mach Number

n_e Number of Elements

n_s Nodal Basis Point

$\mathcal{V}_h^P$ Element Polynomial

p Pressure

p_s Polynomial Degree

q_H Additional Heat Source

R Radius

R Gas Constant

$\mathcal{R}$ Intermediate Runge Kutta Step

S Source Term

S_e Smoothness Indicator

$\vec{s}^*$ Shock Sensor

T Temperature

t Time

U_j Jet Velocity

u Conserved Variable

v_h Test Function

$\mathbf{v}$ Velocity

$\boldsymbol{x}$ Position Vector

Greek Symbols

α User Defined Ramp-up value

β User Defined Kick-in Value

γ Specific Heat Coefficient

δ_{ij} Kronecker Delta

$\boldsymbol{\delta}_i$ Element Correction Field

ϵ Artificial Viscosity

κ User Defined Maximum Value

κ_h User Defined Value

λ Eigenvalue of Semidiscrete Operator

mu Dynamic Viscosity of the Fluid

ξ Mapping Space

ρ Density

$\overline{\overline{\sigma}}$ Stress Tensor

τ Viscous Shear Stress

ϕ Nodal Basis Function

ψ_i Basis Function

Ω Computation Domain

∇ Divergence Operator

Non Alphabetical Symbols

$\|\cdot\|$ Vector Magnitude

$max\,()$ Maximum Value

$min\,()$ Minimum Value

$\otimes$ Tensor Product

Chapter 1

Introduction

Numerical algorithms have been used to solve various mathematical and engineering problems for many years [1]. Preliminary methods were developed for solving simple cases such as linear interpolation but over time methods evolved and more difficult problems started being resolved using new more complex methods. Real world applications of mathematics quickly started to require much more sophisticated algorithms which caught the attention of astronomers and applied mathematicians which is why many of these methods bear their names. As computational power evolved so did the methods used for numerically solving more intricate real world cases. A field in which numerical algorithms are used heavily is computational fluid dynamics (CFD) which relies on a wide array of numerical techniques to solve the partial differential equations that govern the movement of fluids. Although the techniques available to numerically analyze equations with no analytical solution exists they still have certain limitations and different scenarios in fluid movement can strongly influence a solver's ability to capture and resolve the flow. The development of algorithms capable of capturing viscous interactions between fluids and objects, unpredictable chaotic turbulent movement of fluids, fluids moving faster then the speed of sound or even fluids in which chemical reactions are occurring thus releasing energy and changing the properties of the fluid are particularly interesting because of the complexity of these phenomena. By developing numerical solvers capable of capturing such complex physical interactions it allows for more creative and complex engineering solutions to real world problems to be developed and tested in more cost effective ways. More specifically, it is possible to test and validate preliminary ideas without having to build physical prototypes. Many iterations on a design can be tested and perfected at the same time before anything has to be physically built for further testing and validation.

In the context of fluid dynamics, prior to the 1950s, no numerical approach existed for solving complex interactions between different types of flow conditions [2]. The primary motivation for developing methods capable of such interactions was the study of blunt body re-entry vehicles in the context of the space race, which was the source of much of the innovation at the time. The need to find methods capable of transitioning between the subsonic and supersonic

portions of the flow around the re-entry vehicle, as noted in [3], drove Moretti and Abbett to develop a finite difference solution to the problem at hand in [4] which is credited for being the first pragmatic technical solution for such a problem. Understandably, after such a powerful computational tool was developed many more advanced and capable techniques started to emerge. Ironically now this revolutionary method of solving something that helped people reach celestial bodies is a trivial problem which can easily be resolved by students. Currently in industry the most commonly used techniques for solving complex fluid flows are finite volume (FV) methods, mainly because of the robustness and accuracy of the schemes [5]. These schemes do have their shortcomings though in order to obtain higher order accuracy the stencils used increase significantly in complexity and require an increase in spacial discretization, which becomes much more computationally expensive. The need for more compact higher order schemes capable of the same robustness becomes evident in light of these shortcomings for the FV method.

One of the family of methods capable of higher order accuracy using a compact stencil is known as the discontinuous galerkin (DG) methods. This revolutionary approach in CFD contrary to the FV method uses higher order polynomials within the single cell for higher order which means that schemes doesn't require the same spacial discretization as other methods. Similar to finite volume methods the DG method using flux reconstruction (FR) is capable of handling discontinuities at the boundaries of elements [6] meaning that continuity is not required between elements which gives this method many other attractive features. This locally conservative method is capable of handling complicated mesh geometries along with adaptively refined regions with hanging nodes, it is also possible to have polynomials of different degree in adjacent elements meaning two adaptive strategies are available [7]. DG was first introduced in 1973 by Reed and Hill but not in the context of CFD instead it was used for modeling neutron transport [8, 9], since then it has gained a great deal of popularity and has been used to solve a wide range problems relating to gas dynamics, compressible and incompressible flow along with many other applications relating to fields other than CFD [7]. Like all other methods in CFD, DG is not flawless. It has shortcomings, because of the higher order polynomial present in each element. Continuity is enforced within the element, meaning that if a shock is present inside said element, the function used to describe the conserved variables needs to handle that discontinuity. When sharp jumps are present, a phenomenon occurs where unnatural oscillations arise that can cause the solution to give nonphysical results. This phenomenon, known as Gibbs oscillations, needs to be mitigated in order to avoid causing the method to give results that would be physically impossible. Neglecting this phenomenon causes the solution obtained to be inacurate [10, 11].

DG methods use a continuous high order polynomial within each element in order to achieve higher order accuracy using a compact stencil [5]. When discontinuities are present within the domain they usually occure inside an element. This means that the continuous function used to describe the conserved variables, may exhibit undesirable oscillations which could cause the method to blow up. In order to diminish this negative behaviour, various methods are employed. Among these methods is the introduction of artificial viscosity in the conservation equation [12, 13, 14]. In the context of CFD problems in transonic or supersonic flows shocks will cause these undesirable oscillations. If artificial viscosity

is added uniformly in the domain this will inevitably introduce errors and render the results obtained from simulations useless. In order to minimize the amount of artificial viscosity added and remove the unwanted oscillations, techniques which determine where shocks are located in a domain and how much viscosity to add are employed, thus making the DG method one capable of handling complex interactions between shocks, discontinuities and turbulent regions.

Other effective shock detection and stabilization techniques are used in DG, most notably limiting type methods. One family of limiters is based on the methods used in second-order finite volume methods. The methods essentially compares the slope of the solution in the current cell with that in adjacent cells. Although these types of methods are effective and highly used, they have their shortcomings. They degrade smooth regions of the solution and have difficulty with unstructured meshes. An alternative approach is to refine the polynomial degree, typically associated with the higher order polynomial in order to stabilize the solution in the presence of shocks. One of the main limitations of this method is the computational cost associated with using them. In addition to various methods used to stabilize the oscillatory behavior associated with discontinuities in the solution various methods are used for detecting such regions in the flow. Some methods compare the slope and features of the solution with neighboring elements while others rely on the superconvergence property of the discontinuous Galerkin methods. Additional information on the various methods used to detect shocks and handle oscillatory behavior around them can be found in [15, 16, 17, 18, 19, 20, 21].

Alternatively a filtering approach can be utilized to remove unwanted behaviors from the solution. Such an approach attempts to reduce the amount of instability transmitted to the rest of the domain at the edge of discontinuities. This has a similar effect to the use of limiters and artificial viscosity which are active in regions where discontinuities are present and decay in smooth parts of the solution. The danger with filtering, similarly to all other methods of shock mitigation, is that if the filter is too week or too strong the overall solution is negatively affected. Ferrero and Larocca [22] developed a feedback filtering approach based on [23] which attempts to filter out higher order modal coefficients responsible with oscillations as well as their time derivatives using feedback filtering procedure explained in [24]. One problem relating to the filtering method for handling oscillations at points of discontinuity is what type of filter, how often and when the best time to apply the filter is. One advantage of this method is that it amounts to a matrix-vector multiplication which requires little computational resources. A wide range of filtering techniques and approaches can be further investigated in [25, 26, 27, 28].

In this work, the effectiveness of different shock capturing methods at identifying flow discontinuities (notably shock waves) is investigated. These detection methods are then used to implement artificial viscosity and regularize discontinuous solutions using the High-ORder Unstructured Solver (HORUS) algorithm in the discontinuous galerkin/flux reconstruction CFD framework. In chapter 2, a description of the DG method along with the temporal scheme used is reviewed. Chapter 3, the numerical improvements made to HORUS are described and the shock capturing and artificial viscosity methods investigated are described. In chapter 4, benchmark cases in one and two dimensions are investigate to determine the solvers capabilities. Chapter 5 evaluates the feasibility of using a polynomial adaptive approach for

increasing accuracy. Chapter 6 compares results of a real world laboratory experiment with results obtained using the numerical solver. Finally, chapter 7 concludes the work and suggest areas which should be further investigated.

Chapter 2

Theory

In order to obtain higher order accuracy while using a compact stencil, on complex geometries, methods other than the conventional FV method currently used in industry need to be employed. It is for this reason that the DG method is adopted using FR allowing for the solver to utilize a compact stencil capable of higher order accuracy while handling discontinuities between elements with ease. Contrary to FV methods because the DG method uses a high order polynomial within the element, discontinuities can be difficult to evaluate and cause unwanted oscillations known as Gibbs phenomenon. In order for the method to be robust and reliable, a mechanism for mitigating these oscillations is required Artificial viscosity combined with shock detection has been shown to be capable of doing just that in a consistent manner. The theory behind the DG method and the temporal scheme implemented within the HORUS framework are describe in this section. The methodology developed and researched for tracking and controlling shocks do not affect the FR DG scheme, context is required to fully appreciate the effects on the algorithms behavior.

2.1 Conservation Equations

Consider a fluid which occupies a fixed region in space denoted by Ω, confined by a surface $\partial\Omega$. The conserved variables u contained within this region of fluid are affected by source terms $S(u,t)$ and the flow of u in and out of the surface $\partial\Omega$, ie. the flux F. The standard form of the conservation equation can be written in integral form as

$$\frac{\partial}{\partial t}\int_{\Omega} u(t)d\Omega + \int_{\partial\Omega} \boldsymbol{F}(u)\cdot d\boldsymbol{s} = \int_{\Omega} S(u,t)d\Omega. \tag{2.1}$$

In this form ds represents the outward facing vector of the surface confining the element $\partial\Omega$. Through use of Gauss' theorem, the equation is written in the more commonly used divergence form as follows

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{F} = S(u, t). \tag{2.2}$$

2.2 Governing Equations of Compressible Flow

The conservation equations are a mathematical formulation stating that a measurable property of an isolated system doesn't change over time. In laymen terms things are neither created nor destroyed; they move from one place to another as time passes. This is described using conservation equations for mass, momentum and energy. The conservation of mass equation describes how mass moves through a relationship between the density of the fluid in an element and its velocity. In this case, the conserved variable $u = \rho$ and the integral form of the equation takes the following form

$$\frac{d}{dt}\int_{\Omega} \rho d\Omega + \int_{\partial\Omega} \rho(\mathbf{v} \cdot d\boldsymbol{s}) = 0. \tag{2.3}$$

Use of the Gauss theorem gives the divergence form

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \mathbf{v}) = 0 \tag{2.4}$$

Momentum is the product of the mass and velocity vector, and is subject to Newton's second law. In this case, the conserved variable is a vector meaning it has a magnitude as well as a direction. The conserved value is $\boldsymbol{u} = \rho\mathbf{v}$. It is important to also consider internal and external forces acting on the element in any possible direction. The equation in integral form for the conservation of momentum is

$$\frac{d}{dt}\int_{\Omega} \rho \mathbf{v} d\Omega + \int_{\partial\Omega} \rho \mathbf{v}(\mathbf{v} \cdot s) = \int_{\Omega} \rho \boldsymbol{f}_e d\Omega + \int_{\partial\Omega} \overline{\overline{\sigma}} \cdot d\boldsymbol{s}, \tag{2.5}$$

where $\boldsymbol{f}_e$ is the body force and $\overline{\overline{\sigma}}$ is the stress tensor written as

$$\overline{\overline{\sigma}} = -p\overline{\overline{\boldsymbol{I}}} + \left[\left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j}\right) - \tilde{\lambda}(\boldsymbol{\nabla} \cdot \mathbf{v})\delta_{ij}\right]. \tag{2.6}$$

In the stress tensor δ_{ij} is the Kronecker delta and the dilation term being applied to normal stresses is represented by $\tilde{\lambda}(\boldsymbol{\nabla} \cdot \mathbf{v})$, with a typical value of $\tilde{\lambda} = 2/3$ used for Newtonian fluids. Through the use of Gauss' theorem, the divergence

form of the conservation of momentum is

$$\frac{\partial(\rho \mathbf{v})}{\partial t} + \boldsymbol{\nabla} \cdot \left(\rho \mathbf{v} \otimes \mathbf{v} - p\overline{\overline{\boldsymbol{I}}} - \left[\left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j} \right) - \tilde{\lambda} (\boldsymbol{\nabla} \cdot \mathbf{v}) \delta_{ij} \right] \right) = 0. \tag{2.7}$$

Lastly, through the application of the first law of thermodynamics on a closed system, the law of conservation of energy is derived. In this case, the conserved variable is $u = \rho E$, the convective flux is represented by $\boldsymbol{F}_C = \rho \mathbf{v} E$ and the diffusive flux, written using Fourier's law of heat conduction, is $\boldsymbol{F}_D = -k \boldsymbol{\nabla} T$. Here k represents the thermal conductivity, and T is the temperature. Alternative sources of energy to be considered are work, viscous shear, and pressure representing internal forces through the stress tensor $\overline{\overline{\sigma}}$. Additional heat sources are represented by q_H. The energy conservation equation is thus written as

$$\frac{d}{dt} \int_\Omega \rho E d\Omega + \int_{\partial\Omega} (\rho E \mathbf{v} - k \boldsymbol{\nabla} T) \cdot ds = \int_\Omega (\rho \boldsymbol{f}_e \cdot \mathbf{v} + q_H) b\Omega + \int_{\partial\Omega} \overline{\overline{\sigma}} \cdot \mathbf{v} d\boldsymbol{s}, \tag{2.8}$$

where $\overline{\overline{\sigma}}$ is the stress tensor defined using equation 2.6 The divergence form of the above equation is

$$\frac{\partial(\rho E)}{\partial t} + \boldsymbol{\nabla} \cdot \left(\rho \mathbf{v} H - k \boldsymbol{\nabla} T - \left[\left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j} \right) - \tilde{\lambda} (\boldsymbol{\nabla} \cdot \mathbf{v}) \delta_{ij} \right] \cdot \mathbf{v} \right) = \rho \boldsymbol{f}_e \cdot \mathbf{v} + q_H, \tag{2.9}$$

where $W_f = \rho \boldsymbol{f}_e \cdot \mathbf{v}$ represents the work from body forces and H represents the stagnation enthalpy defined as

$$H = e + \frac{p}{\rho} + \|\mathbf{v}\|^2 = \mathrm{h} + \|\mathbf{v}^2\| = E + \frac{p}{\rho}. \tag{2.10}$$

Pressure is represented by variable p and $\|\cdot\|$ is the magnitude of a given vector. The ideal gas law, $p = \rho R T$ is used, with R as the specific gas constant and temperature represented by T. The temperature relationship between internal energy e and enthalpy h is that of a perfect gas

$$e = c_v T, \tag{2.11}$$

$$\mathrm{h} = c_p T. \tag{2.12}$$

The specific heat coefficient for constants at volume and pressure being c_v and c_p respectively. The specific heat ratio $\gamma = c_p / c_v$ relates the specific heats. A value of $\gamma = 1.4$ is used for air. Pressure, following the ideal gas law can be expressed as the function of energy

$$p = (\gamma - 1) \rho \left(E - \frac{1}{2} \|\mathbf{v}\|^2 \right). \tag{2.13}$$

2.3 Euler and Navier-Stokes Equations

The conservation laws above are specialized to the Euler and Navier-Stokes equations. The fluxes used are $\boldsymbol{F}_C$ and $\boldsymbol{F}_D$ which represent the transport flux and diffusive flux respectively.

$$\frac{\partial \boldsymbol{u}}{\partial t} + \overline{\overline{\boldsymbol{\nabla}}} \cdot \left(\overline{\overline{\boldsymbol{F}}}_C - \overline{\overline{\boldsymbol{F}}}_D \right) = \boldsymbol{S}. \tag{2.14}$$

The divergence form of the given equations can be written in the vector notation which is most commonly used in CFD solvers as

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial}{\partial x}(\boldsymbol{F}_{C,x} - \boldsymbol{F}_{D,x}) + \frac{\partial}{\partial y}(\boldsymbol{F}_{C,y} - \boldsymbol{F}_{D,y}) + \frac{\partial}{\partial z}(\boldsymbol{F}_{C,z} - \boldsymbol{F}_{D,z}) = \boldsymbol{S}, \tag{2.15}$$

where $\boldsymbol{u}$ is the vector of conserved variables

$$\boldsymbol{u} = \begin{bmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ \rho E \end{bmatrix}. \tag{2.16}$$

The difference between the Euler and Navier-Stokes are observed in the fluxes used in the conservation equations. For the Euler equations, the effects of viscosity and heat transfer along with the source term are considered to be negligible. This means that the only flux to be considered is the convective flux

$$\boldsymbol{F}_{C,i} = \begin{bmatrix} \rho v_i \\ \rho v_i v_x + \delta_{ix} p \\ \rho v_i v_y + \delta_{iy} p \\ \rho v_i v_z + \delta_{iz} p \\ \rho v_i H \end{bmatrix}, \tag{2.17}$$

where the spatial coordinates are represented by $i = [x, y, z]$, and δ_{ij} is the Kronecker delta.

On the other hand in the case of the Navier-Stokes equation the effects of viscosity, heat conduction and source terms are not ignored. The Navier-Stokes equation gives a more complete and realistic picture of how viscous fluids behave necessary for simulating turbulent flow. The addition of these new effects gives a diffusive flux and source term

that are formulated as follows

$$\boldsymbol{F}_{D,i} = \begin{bmatrix} 0 \\ \tau_{xi} \\ \tau_{yi} \\ \tau_{zi} \\ \boldsymbol{\tau}_i \cdot \mathbf{v} + k\partial_i T \end{bmatrix}, \tag{2.18}$$

$$\boldsymbol{S} = \begin{bmatrix} 0 \\ \rho f_x \\ \rho f_y \\ \rho f_z \\ \rho \boldsymbol{f}_e \cdot \mathbf{v} + q_H \end{bmatrix}, \tag{2.19}$$

where τ is the shear stress due to viscosity, f_i and $\partial_i T$ are the body forces and partial derivatives of temperature in the various coordinates such that $i = [x, y, z]$.

2.4 Discontinuous Galerkin Method

The ability for the DG method to obtain higher order accuracy while maintaining a compact stencil stems from its utilization of higher order polynomials within each element to describe the conserved variable [7, 29]. Since the continuity of the polynomial is only enforced within the element jumps between elements are handle using a method similar to the one used for FV methods to resolve the inconsistencies in the flux at the boundary. The DG method starts with equation 2.2 in which $S(u,t)$ is considered equal to zero. The domain Ω is split into elements $n_e \in T_h$ an approximate solution denoted u_h in a space of element-wise polynomials

$$\mathcal{V}_h^P = \{\boldsymbol{v} \in \boldsymbol{L}^2(\Omega) : \boldsymbol{v}|_{n_e} \in \boldsymbol{P}^p(n_e) \forall n_e \in \boldsymbol{T}_h\}. \tag{2.20}$$

By integrating over the discretized domain n_e in equation 2.2 and multiplying by the test function $\boldsymbol{v}_h \in \mathcal{V}_h^p$ taken from the same space the following integral is obtained

$$\int_{n_e} \left[(\boldsymbol{u}_h)_t + \nabla \cdot F_i(\boldsymbol{u}_h)\right] \boldsymbol{v}_h d\boldsymbol{x} = 0. \tag{2.21}$$

After integrating by parts, the integral becomes

$$\int_{n_e} \left[(\boldsymbol{u}_h)_t\right] \boldsymbol{v}_h d\boldsymbol{x} - \int_{n_e} F_i(\boldsymbol{u}_h) \nabla \boldsymbol{v}_h d\boldsymbol{x} + \int_{\partial n_e} \hat{F}_i\left(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \hat{\boldsymbol{n}}\right) \boldsymbol{v}_h^+ ds = 0. \tag{2.22}$$

After this integration by parts, a flux contribution appears over the boundary of the element, $\hat{F}_i(\boldsymbol{u}_L, \boldsymbol{u}_R, \hat{\boldsymbol{n}})$, meaning fluxes are not only the boundary of the domain. The beauty with this contribution is that continuity is not enforced between elements and there is a class of methods capable of handling these discontinuities. The only continuity in forced is on the flux between elements. The problem at hand is identical to the one faced when using FV methods, this means that any method which can be used in a FV solver to solve the Riemman problem could be used in the context of the DG method. The global problem is to find a test function $\boldsymbol{u}_h \in \mathcal{V}_h^p$ so that the residual are zero for all $\boldsymbol{v}_h \in \mathcal{V}_h^p$.

2.5 Spatial Discretization

In order to obtain higher orders of accuracy for Euler and Navier-Stokes equation the spatial operator for the general advection-diffusion equations is discretized using a high-order unstructured FR approach which has been implemented in HORUS. Based on the formulation of FR, the solution for each element is represented using a discrete polynomial of degree p such that the conserved variable takes the form [6, 30]

$$\boldsymbol{u}(\boldsymbol{x}, t) \approx \boldsymbol{u}^h(\boldsymbol{x}, t) = \bigoplus_{i=1}^{n_e} \boldsymbol{u}_i^h(\boldsymbol{x}, t). \tag{2.23}$$

The piece-wise continuous approximation for the global solution is represented by $\boldsymbol{u}^h(\boldsymbol{x}, t)$ and the continuous representation for a single element n_e in the domain is given by $\boldsymbol{u}_i^h(\boldsymbol{x}, t)$. The standard nodal basis form is used to approximate the solution in each element as follows

$$\boldsymbol{u}_i^h(\boldsymbol{x}, t) = \sum_{j=1}^{n_s} \boldsymbol{u}_{i,j}(t) \phi_{s,i,j}(\boldsymbol{x}). \tag{2.24}$$

The solution at one nodal basis point, n_s, is given as $\boldsymbol{u}_{i,j}(t)$ for a given element, while $\phi_{s,i,j}(\boldsymbol{x})$ represents the corresponding nodal basis function. This technique enforces continuity within a given element without the need for the solution to be continuous between elements at the boundary. This results in the presence of discontinuities at boundaries and the need to resolve said discontinuities using an appropriate method [6]. Through use of the FR method and expanding it to simplex element types [30, 31] the conservation laws being resolved must be satisfied discretely on each element as follows

$$\frac{\partial \boldsymbol{u}_i^h}{\partial t} + \nabla \cdot \boldsymbol{F}_i^h + \boldsymbol{\delta}_i = 0. \tag{2.25}$$

The polynomial reconstruction of the flux on an elementwise basis is of the form $\boldsymbol{F}_i^h = \boldsymbol{F}_i^h(\boldsymbol{u}_i^h, \nabla \boldsymbol{u}_i^h)$ with $\boldsymbol{\delta}_i$ representing a correction field withing the element that lies in the same polynomial space as the solution. The correction field is comparable to the divergence of the penalty functions proposed in the original FR scheme for tensor product elements [6]. This method enforces continuity of the flux at the boundary between elements without the need for the adjacent elements to require continuous solutions at the element boundaries. When the conservation law is applied at each of the points

that make up the solution in an element it takes the form

$$\frac{d\boldsymbol{u}_{i,j}^{h}}{dt} + \left(\nabla \cdot \boldsymbol{F}_{i}^{h}\right)\Big|_{\boldsymbol{x}_{i,j}} + \boldsymbol{\delta}_{i,j} = 0. \tag{2.26}$$

By basing this on the FR formulation the equation becomes

$$\boldsymbol{\delta}_{i,j} = \frac{1}{|\Omega_i|} \sum_{f \in S} \sum_{l} \alpha_{i,j,f,l} [\hat{\boldsymbol{F}}]_{i,f,l} S_f, \tag{2.27}$$

where $|\Omega_i|$ represents the volume of the i-th element, f denotes the number of one of the faces of the element that make up the elements surface S, the flux points on the surface of the element are represented by l and $\alpha_{i,j,f,l}$ is a constant value known as the lifting coefficient. The common Riemann flux at the flux point and the internal flux value difference is $[\hat{\boldsymbol{F}}]_{i,f,l}$. In order to obtain a range of energy stable schemes for general element types, the value of the lifting coefficient can be varied which in turn yields schemes such as the Spectral Difference (SD), Spectral Volume (SV) and Discontinuous Galerkin (DG) methods. HORUS uses lifting coefficients derived from the nodal basis functions, recovering the DG method as its operating scheme [6, 30]. Following a previously proposed approach for FR and DG, the reference elements are mapped with coordinates $\boldsymbol{\xi}$ for both the solution and discrete system of governing equations [6, 30]. The one-to-one mapping follows $\boldsymbol{x} = M(\boldsymbol{\xi})$ such that $\boldsymbol{\xi} = M^{-1}(\boldsymbol{x})$, a nodal polynomial representation of degree p_g of the mapping function M is defined using the mapping points such that

$$\boldsymbol{x}_{i}^{h}(\boldsymbol{\xi}, t) = \sum_{j}^{n_g} \boldsymbol{x}_{i,j} \phi_{g,i,j}(\boldsymbol{\xi}). \tag{2.28}$$

The interpolated actual location, $\boldsymbol{x}_{i}^{h}(\boldsymbol{\xi}, t)$, is determined through the number of mapping points, n_g, based on the polynomial degree, with $\boldsymbol{x}_{i,j}(t)$ representing the physical location of the mapping points at an instant in time. For any of the mapping points, the determinant can be found with

$$J = \left|\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right|. \tag{2.29}$$

The above change allows for all operations to be performed on the idealized reference element and then converted back to the physical element using the mapping function. Simply put, this represents the change in volume of the local space as it is tranformed to the physical space [32].

2.6 Time Discretization

In order to move equation 2.2 in time after it has been spatially resolved an explicit time stepping method is used. The majority of the problems solved in the context of this research are transient in nature and have no steady state solution. In light of the speed at which things are changing within the simulation and in order to minimize computational costs an explicit solver is used. The time stepping method used is the classical $RK_{4,4}$ which is extensively used in CFD because of the 4th order accuracy it offers. Higher order RK methods are also available but they require additional computational resources. The high order of accuracy and the minimal use of resources required to operate the $RK_{4,4}$ method, described as a Butcher tableau in equation 2.30, make the method especially attractive.

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} . \tag{2.30}$$

Between every intermediate time step the spatial discretization routine is executed as part of the temporal scheme described in equation 2.31

$$\begin{aligned} 1^{st}Stage \longrightarrow & \begin{cases} u_1 = u^t \\ \mathcal{R}_1 = \mathcal{R}(u_1) \end{cases} \\ 2^{nd}Stage \longrightarrow & \begin{cases} u_2 = u^t + \frac{\Delta t}{2}\mathcal{R}_1 \\ \mathcal{R}_2 = \mathcal{R}(u_2) \end{cases} \\ 3^{rd}Stage \longrightarrow & \begin{cases} u_3 = u^t + \frac{\Delta t}{2}\mathcal{R}_2 \\ \mathcal{R}_3 = \mathcal{R}(u_3) \end{cases} \\ 4^{th}Stage \longrightarrow & \begin{cases} u_4 = u^t + \Delta t\mathcal{R}_3 \\ \mathcal{R}_4 = \mathcal{R}(u_4) \end{cases} \\ u^{t+1} = u^t + \Delta t & \left(\frac{1}{6}\mathcal{R}_1 + \frac{1}{3}\mathcal{R}_2 + \frac{1}{3}\mathcal{R}_3 + \frac{1}{6}\mathcal{R}_4\right) \end{aligned} . \tag{2.31}$$

Important considerations need to be taken so that the time step Δt is not so big that it will cause the solution to diverge and give incorrect values. In order to use the appropriate Δt value so that the solution remains stabl,e an adaptive time

stepping routine described in equation 3.4 based on the Courant-Friedrichs-Lewy (CFL) condition is used to find the required Δt value for the solution to remain stable.

Chapter 3

Improvements to HORUS for Shock Capturing

The complex nature of fluid flow means that a wide range of different phenomena can occur, depending on initial conditions, geometry and other underlying properties of the problem at hand. In problems involving fluids moving at high velocities, shocks can arise which present a particular challenge for the numerical solver. The DG method is capable of handling discontinuities at the boundaries but the polynomial within the element is continuous, meaning that discontinuities can cause the solution to act in an unexpected manner potentially leading to inaccurate solutions potentially causing the solver to crash. To circumvent this issue, regions where shocks are present need to be determined and the unwanted oscillations that ensue in these regions need to be controlled to avoid deteriorating the solution. Additionally, since the flow is in a state of constant change, the maximum allowable time step which ensures stability is also changing. In order to determine the maximum allowable time step, the CFL condition is used, allowing for an explicit adaptive time stepping routine to be utilized.

3.1 Courant-Friedrichs-Lewy Condition

In order to accelerate computation while maintaining stability and accuracy, it is important to find the maximum allowable time step that can be taken for a given problem. Using the Courant-Friedrichs-Lewy (CFL) condition, it is possible to find this value with minimal computational requirements. In order to determine its value, a purely advective, one-dimensional case is considered for a given conserved value and the linear advection equation is recovered

$$\frac{\partial u}{\partial t} = c\frac{\partial u}{\partial x}, \tag{3.1}$$

where c is the velocity at which the conserved value u is moving through space. In order to capture the change of the considered value as it travels from one element to the next, it is important that the time steps taken are smaller than the time it takes to travel through a single element. If time steps larger than this value are used, information about interactions with intermediate elements is lost and the solution becomes unstable. By imposing this condition the maximum allowable time step used follows

$$\Delta t \leq \frac{\Delta x}{c}. \tag{3.2}$$

This is true for an idealized, one dimensional problem. In the case of multidimensional problems, the direction of travel, and distance between mapping points of the elements need to be taken into account. Additionally, since the element solution is made up of a higher order polynomial with a given set of nodal points, the solution is not constant throughout the element, adding an additional challenge. In order to solve for the maximum allowable time step for cases with more than one spatial dimension two approaches are taken. The first follows the following form

$$\Delta t = CFL\left(\frac{\min(\|\boldsymbol{x}_{ij}\|)}{\max(\|\mathbf{v}_k\|) + \max(C_m)}\right), \tag{3.3}$$

where C_m represents the speed of sound in a given elements nodal point m, calculated using $C = \sqrt{\gamma p/\rho}$. The vector $\boldsymbol{x}_{ij}$ represents the distance between different mapping points for the element and $\mathbf{v}_k$ is the velocity at a given point. In this case the values obtained for the various parameters don't need to come from the same nodal point which means that the direction of the fastest wave and shortest distance may be perpendicular, resulting in a relationship between the two values which is not one of interest for calculating the maximum allowable time step. If this is the case, the resulting value of Δt used can be smaller than required to remain stable. By cycling through all the elements in the computational domain a global time step can be determined and used for the explicit time stepping technique shown in equation 2.31. The second approach is formulated to circumvent this issue

$$\Delta t = \min\left(\frac{\|\boldsymbol{x}_{ij}\|}{\mathbf{v}_i \cdot \frac{\boldsymbol{x}_{ij}}{\|\boldsymbol{x}_{ij}\|} + C_i}\right) CFL, \tag{3.4}$$

where the velocity and sound speed used are only evaluated at point i as the dot product of the velocity and directional unit vector are tested for other mapping points to find the maximum allowable time step. The value of $i \neq j$ and i cycles through all mapping points of the element. This approach allows for potentially longer time steps to be taken, by calculating the time for a signal (eg a sound wave) to travel across a single cell and its local speed and along a the direction considered. In the previous method, the maximum velocity vector may not have been oriented in the direction of the shortest cell dimension. This mismatch between the wave speed and cell dimension orientation could lead to an underestimation of the maximum allowable time step.

One pitfall of both approaches is that they only hold true for linear advective problems. If non-linear or diffusive

processes are present, it is possible that the time step calculated does not satisfy the stability criteria for a given scenario. For this reason the value of the CFL number must be kept low to ensure solution convergence. Additionally, the introduction of artificial viscosity to dampen the solution in regions where shocks are present adds further complexity to the problem and canfurther influence the maximum allowable time step. Alternative approaches for calculating the maximum allowable values of Δt based on the CFL condition have been proposed in [33, 34].

3.2 Shock Capturing and Artificial Viscosity

The use of higher order methods may have many advantages but it also has some drawbacks. One of these drawbacks is that when the flow involves steep slopes caused by the presence of shocks or contact discontinuities, the solution can become unstable. These instabilities are known as Gibbs phenomenon [10, 11] which occur when large oscillations are present in the higher order function. These oscillations create an unrealistic solution which can give negative pressure or density potentially cause the solution to blow-up.

The Gibbs phenomenon can be contained so that the solution converges. This can be done in several ways but the methodology focused on is the addition of artificial viscosity that will dampen the oscillations without skewing the solution. The strength and location where artificial viscosity is added must be chosen carefully. One approach would be to add viscosity everywhere in the domain, but this would cause inaccuracies in the solution since the viscosity term is not present in the original governing equation. To circumvent this issue, a shock capturing sensor is developed which will allow the artificial viscosity to be added only in regions where there are large jumps in the solution. In addition to ensuring that the viscosity is only added in the cells that require damping, due to unnatural oscillations, the amount of damping added needs to be controlled. This is done by selecting a suitable function that will add viscosity according to the level of oscillations in the solutions. This ensures no excessive amount of viscosity is added in different cells, subsequently minimizing the effect of the added term on the solution.

3.2.1 Smoothness Shock Capturing

Persson and Peraire [14] developed an approach for the addition of artificial viscosity which is based on the smoothness of the density. This sensor is designed to determine cells in which the density distribution is increasing or decreasing at a discontinuous rate which would indicate the presence of a shock. Subsequently, the viscosity function will add large amounts of viscosity in such areas and quickly reduce the amount of viscosity added in regions considered smooth by the sensor. In order to do this, a dissipative model term is added to the conservation law presented in equation 2.2 for which the source term has been set to zero. The modified equation then becomes

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{F} = \nabla \cdot (\epsilon \nabla u). \tag{3.5}$$

The smoothness sensor, designed to determine regions where artificial viscosity is required, does so by writing the solution for each element in a hierarchic grouping of orthogonal polynomials, the solution of order p in each element is expressed in an orthogonal basis as follows

$$\boldsymbol{u} = \sum_{i=1}^{N(p_s)} u_i \psi_i, \tag{3.6}$$

where $N(p_s)$ is the total number of terms and ψ_i is the basis function. When the solution is not smooth, the strength of the discontinuity will determine the rate at which the expansion coefficient will die down. An additional truncated term with the same solution is also considered but containing only $p_s - 1$ terms

$$\hat{\boldsymbol{u}} = \sum_{i=1}^{N(p_s-1)} u_i \psi_i. \tag{3.7}$$

Each element Ω_e has the smoothness indicator defined as

$$S_e = \frac{(u - \hat{u}, u - \hat{u})_e}{(u, u)_e}, \tag{3.8}$$

with $(.,.)_e$ being the standard inner product of $L_2(\Omega_e)$. If a shock is detected using the above indicator, viscosity is added as a constant in a given element using the smooth function

$$\epsilon_e = \begin{cases} 0, & s_e \leq s_0 - \kappa, \\ \frac{\epsilon_0}{2}(1 + \sin(\frac{\pi(s_e - s_0)}{2\kappa})), & s_0 - \kappa \leq s_e \leq s_0 + \kappa, \cdot \\ \epsilon_0, & s_e > s_0 + \kappa. \end{cases} \tag{3.9}$$

For the above function, $s_e = log_{10} S_e$, $\epsilon_0 \sim h/p$ and $s_0 \sim 1/p^4$. The value of κ is a user defined value that is selected on the basis of obtaining sharp shock profiles in solutions free of oscillations.

3.2.2 Dilation Shock Capturing

An alternative approach to shock detection is a method based on the dilation of velocity. The presence of shocks is strongly correlated with negative dilation making the approach based very promising. Moro et al. [13] used dilation as a means of detecting shocks and applying artificial viscosity to the governing equation in a similar manner as Persson [14] by adding an additional artificial viscosity term as follows

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{F} = \nabla \cdot \epsilon \nabla \boldsymbol{u}_{AV}(\boldsymbol{u}). \tag{3.10}$$

In this case, $\boldsymbol{u}_{AV}(\boldsymbol{u}) = [\rho, \rho\mathbf{v}, \rho\boldsymbol{H}]$. This differs from u presented in equation 2.16 on the last term which ensures conservation of enthalpy across the shock for steady flow and maintaining dissipative properties for transient cases. The amount of viscosity added, ϵ, is a function of the dilation, state of the fluid and the grid resolution. The method used by [13] is made non-dimensional by basing their work on an approach developed by Person and Peraire which takes a characteristic length term of order of h/p with p representing the polynomial degree and h representing the element size. In addition to the length scale needed, a velocity scale is required for the non-dimensional dilation term. Alternatively to previous works which use the speed of sound $c = \sqrt{\gamma RT}$ which is subject to change from pre and post shock areas the critical speed of sound c^* is proposed as an alternative to c.

$$c^* = \sqrt{\gamma RT^*} = \sqrt{\gamma R\left(\frac{2}{\gamma+1}\right)T_0}. \tag{3.11}$$

By using the above length and velocity scales, the shock sensor $\bar{s}^*(\boldsymbol{u})$ is defined:

$$\bar{s}^*(\boldsymbol{u}) = -\frac{(k_h h/p_s)\nabla\cdot\mathbf{v}}{c^*}. \tag{3.12}$$

The artificial viscosity term, ϵ, added to the governing equation is given as

$$\epsilon = \left(k_h \frac{h}{p_s}\right)\sqrt{\mathbf{v}\cdot\mathbf{v} + c^2}\, f(\bar{s}^*). \tag{3.13}$$

The function f is defined as

$$f(x) = \frac{log(1 + exp(\alpha(x-\beta)))}{\alpha}. \tag{3.14}$$

The α term governs the shape of the function or the ramp up of the viscosity while β determines the kick-in of the viscosity. Both are user defined terms. Based on experimental results in [13], if is found that $\alpha = 10^4$ and $\beta = 0.1$ give good solutions.

3.2.3 Modified Shock Capturing

The proposed approach uses a modified form of the dilation-based shock indicator by Moro et al. [13]. The governing equation is taken in the similar form stated above

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla\cdot\boldsymbol{F} = \nabla\cdot(\epsilon\nabla\boldsymbol{u}) \tag{3.15}$$

where $\epsilon = \epsilon(\boldsymbol{u}, \nabla\boldsymbol{u})$ is a Laplacian artificial viscosity. The objective is the same as in the other methods where ϵ is chosen so that enough artificial viscosity to dampen excesive oscillations in the proximity of shocks is added, while

keeping smooth flow regions undisturbed. A non-dimensional shock sensor based on the work from Moro et al. [13] is proposed

$$\tilde{s}^* = -k_h h \frac{\nabla \cdot \mathbf{v}}{c^*}, \tag{3.16}$$

where k_h is a non-dimensional user defined tuning parameter, and h is a length scale. Contrary to Moro et al. [13], who use a piece-wise linear reconstruction using a local length scale based on the smallest altitude in any given element, the length scale used is

$$h = \frac{h_\xi J^{1/n_d}}{p_s + 1}, \tag{3.17}$$

where h_ξ is a characteristic length of the element in reference space, and n_d is the number of physical dimensions. This form of the length scale allows h to vary inside the element based on several variables that may differ between elements, more specifically the polynomial degree and the shape deformation based on the reference element. The same equation as in [13] is taken for the critical sound speed

$$c^* = \sqrt{\frac{2\gamma R T_0}{\gamma + 1}} \tag{3.18}$$

It is important to note that wherever Equation 3.16 is positive, compression is present and likely presence of a shock. The artificial viscosity scale has the same form as [13] with the modified length scale h, such that

$$\epsilon = k_h h \sqrt{\mathbf{v} \cdot \mathbf{v} + c^2} f(\tilde{s}^*), \tag{3.19}$$

The function $f(x)$ represents the activation function for the amount of artificial viscosity added, depending on the dilation based shock sensor which is a modification on the work of Persson and Perair [14]. The new activation function proposed is

$$f(x) = \begin{cases} 0, & x \leq \beta, \\ \frac{\cos(\pi(\alpha(x-\beta)-1))+1}{2}, & \beta < x < 1/\alpha + \beta, \\ 1, & x \geq 1/\alpha + \beta. \end{cases} \tag{3.20}$$

where β is chosen to remove artificial viscosity in smooth regions of the flow, and α determines the rate at which artificial viscosity is added with a maximum value of $k_h h \sqrt{\mathbf{v} \cdot \mathbf{v} + c^2}$. Additional information on various other approaches to adding artificial viscosity within the DG framework can be found in [12].

Chapter 4

One and Two-Dimensional Benchmark Cases

All numerical methods must go through a rigorous validation and testing phase in order to determine their effectiveness at solving a wide range of different types of scenarios. Extensive work has been done on evaluating the best methodology for verifying and validating CFD codes, but no general consensus emerges [35]. In order to determine the capabilities of the HORUS solver, preliminary, 1D convergence tests on smooth problems are performed, ensuring that the solver gives the expected results. Building on this, additional 1D tests with discontinuities are used to determine HORUS' ability to handle shocks and their interactions with complex phenomena. The next natural step is to add an additional spatial dimension and repeat the testing method used for the one dimensional case by comparing to the 1D results obtained.

4.1 Convergence of One-Dimensional Problems with Exact Solutions

In order to determine the effectiveness of the higher order solver, a first set of convergence tests were conducted to determine the order of convergence for varying polynomial degrees. The convergence tests were also used to identify if the shock capturing technique being utilized was giving the same order of convergence for smooth problems, where it is not required. By doing so, it ensures that artificial viscosity is not added in regions where shocks are not present and the solution does not behave in a discontinuous manner.

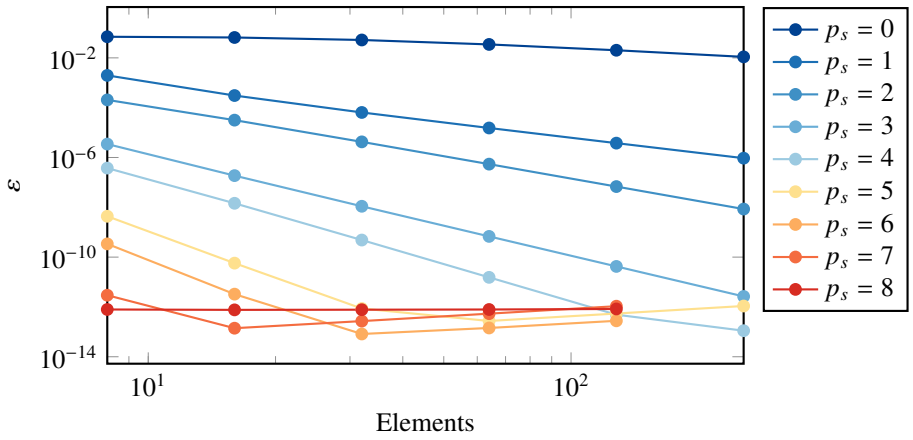


Figure 4.1. Convergence plot of the sine wave density distribution using fixed time stepping and shock capturing turned off.

4.1.1 Density Wave

To verify that the proposed shock capturing scheme can maintain high-order accuracy in smooth flows, we consider the advection of a one-dimensional density wave using the Euler equations. The initial condition is specified as

$$\begin{aligned} \rho &= 1 + 0.1\sin(2\pi x), \\ u &= 1.0, \\ p &= 1.0, \end{aligned} \tag{4.1}$$

and a periodic domain of unit length is used with $n_e = 8$, 16, 32, 64, 128, and 256 elements. The classical, explicit, fourth-order Runge-Kutta scheme described in equation 2.31 is used in time. The conservation equations being solved are the inviscid Euler equations resulting in the wave traveling through the domain with no change in phase or amplitude.

Table 4.1. Rate of convergence for moving density wave for varying computational routines.

p_s	$Fixed\Delta t$	$CFL\Delta t$	$ModifiedDilation$	$Smoothness$
0	0.8833	0.8833	0.8833	0.8833
1	2.6854	2.6855	2.6855	1.5248
2	2.9981	2.9981	2.9981	10.8319
3	4.2113	4.2112	4.2112	15.9525
4	4.9836	4.9828	4.9828	4.9929
5	6.2511	6.2103	6.2103	6.2104
6	6.7010	6.3189	6.3189	6.3146
7	4.3987	4.0089	4.0089	4.0241
8	0.0518	3.2697	3.2697	3.4631

After investigating the error convergence rate for varying polynomial degrees based on grid refinement, an investi-

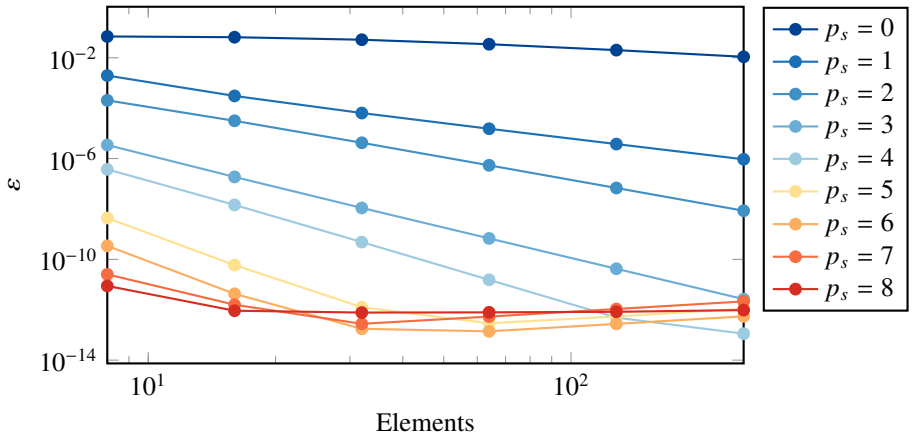


Figure 4.2. Convergence plot of the sine wave density distribution using adaptive time stepping and shock capturing turned off.

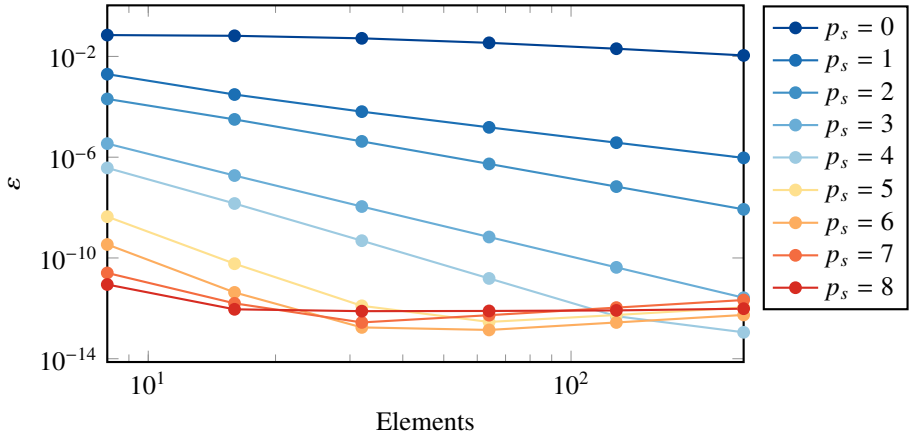


Figure 4.3. Convergence plot of the sine wave density distribution using variable time stepping and dilation based shock capturing turned on.

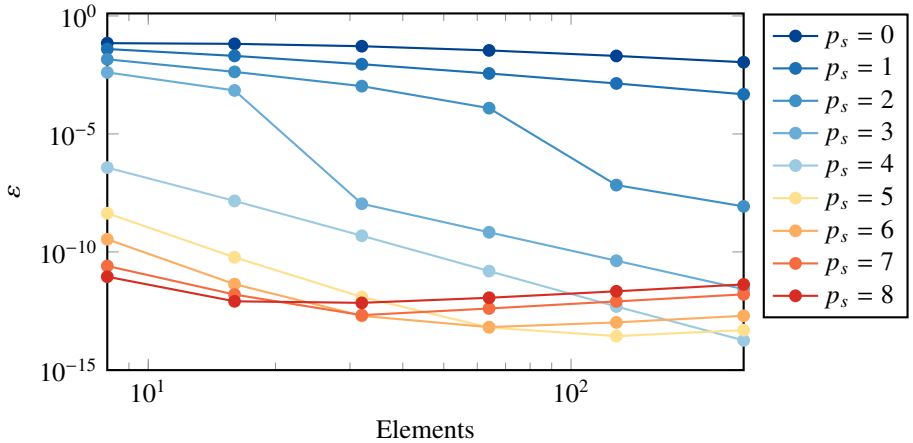


Figure 4.4. Convergence plot of the sine wave density distribution using variable time stepping and smoothness based shock capturing turned on.

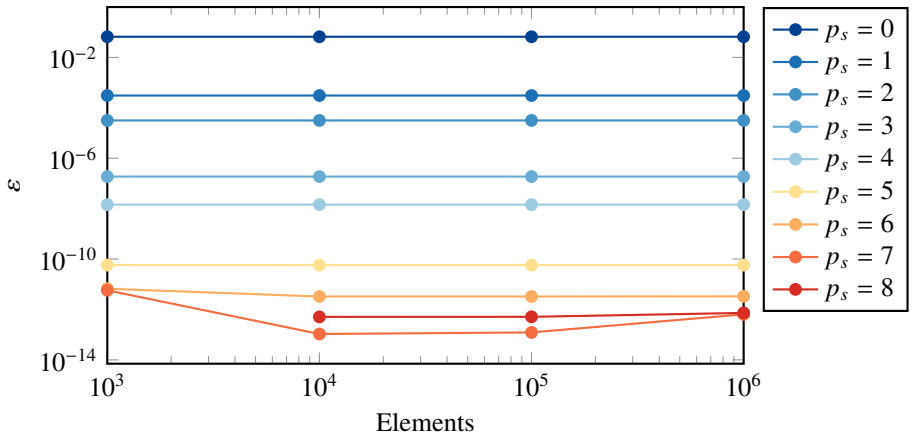


Figure 4.5. $L2$ error based on time convergence on fixed mesh of 16 elements and varying polynomial degrees.

gation into the solvers sensitivity to time step refinement was conducted. Figure 4.5 shows the $L2$ error convergence for varying time steps. This graph shows that the reduction in error obtained by refining the time step is within the range of machine precision on coarser meshes such as the ones investigated in figures 4.1, 4.2, 4.3 and 4.4. This behavior is expected especially for higher order polynomial degrees where the contribution to the error from the spatial scheme is significantly larger than the temporal scheme. In the case of $p_s = 7$ the dip in ϵ can be explained by the fact that the spatial scheme error contribution is now so little that the temporal scheme error dominates before the error contribution from both reduces to the order of machine precision. Further explanation of this phenomenon can be found in [36, 37] where a more in depth analysis of the temporal convergence of explicit time stepping schemes in the DG/flux reconstruction framework are investigated.

For each polynomial degree, a different value of Δt was used and kept constant for each subsequent level of grid refinement results are shown in figure 4.1. In a similar manner, for cases where adaptive time stepping was used, results are plotted in figures 4.2, 4.3 and 4.4, a different value of the CFL number was used for each polynomial degree and kept constant for each level of grid refinement. Contrary to FV methods which typically used CFL values close to 1 as the polynomial degree is increased the CFL number used decreases and becomes significantly smaller than 1. In the context of the moving density wave when the adaptive time stepping routine was used the CFL numbers used were 0.8, 0.3, 0.14, 0.09, 0.06, 0.045, 0.03, 0.026 and 0.02 for polynomials ranging from 0 to 8. It can be observed in table 4.1 that the order of convergence increases as the polynomial degree is increased. This tends to fall appart for $p_s > 6$ most likely because of the accuracy limitations on the temporal scheme which is 4th order accurate. Additionally, it is observed in figure 4.4 that when the smoothness shock capturing is turned on the convergence curves don't follow a linear slope, as would be expected, when plotted on a log-log scale. This is due to the fact that artificial viscosity is being added to elements where no shocks are present, greatly skewing the results. In table 4.1, which shows the largest slop for varying approaches, the jumps in convergence observed in figure 4.4 are presented numerically. For this reason, and

after comparing convergence of the SOD shock tube it was decided to pursue a dilation based shock capturing approach in all future tests with no exact solution.

When comparing the results obtained with both the modified dilation based shock capturing and adaptive time stepping turned on with results obtained with adaptive time stepping turned on but no artificial viscosity no differences in convergence rate are observed. This gives a strong indication that the method is capable of resolving smooth areas of the flow without triggering the shock sensor. This is further confirmed in other test cases.

4.1.2 Sod Shock Tube

In order to choose suitable values of α, β, and k_h, and demonstrate the utility of polynomial adaptation, we consider Sod's shock tube, which is a commonly used test case for validation of numerical methods in supersonic regimes. We use a domain of length $x \in [0, 1]$ split in two parts: one high pressure, high density region to the left, and a low pressure and density region to the right with ratios of 10:1 and 8:1 respectively as described in 4.2. The left and right conditions given by

$$\begin{pmatrix} \rho_L \\ p_L \\ u_L \end{pmatrix} = \begin{pmatrix} 1.0 \\ 1.0 \\ 0.0 \end{pmatrix}, \quad \begin{pmatrix} \rho_R \\ p_R \\ u_R \end{pmatrix} = \begin{pmatrix} 0.125 \\ 0.1 \\ 0.0 \end{pmatrix}, \tag{4.2}$$

where ρ_L, ρ_R, p_L, p_R, u_L, and u_R are the density, pressure, and velocity on the left and ride hand sides of the shock tube interface, which is taken to be at the center of the domain. This simple yet interesting test case offers an analytical solution, which can be found in [38]. This ability to compare the computational results with an exact analytical solution explains why it is a benchmark in the CFD community. These initial conditions are known to form a shock, contact discontinuity, and expansion fan as the solution evolves in time. A final time of $t = 0.2$ was selected for convenience. The solution to the Euler conservation equation is represented on each element Gauss points and the Rusanov scheme is used to compute the common Riemann flux between elements.

It can be observed, in figure 4.6, that as the polynomial degree is increased, the gradient at the shock also increases indicating that the higher order polynomials are capable of producing a sharper change in density, better reflecting the physical solution. Figures 4.7 and 4.8 show the grid convergence plots for the Sod shock tube test case using the modified shock capturing approach which relies on dilation for the detection of shocks and the smoothness shock capturing approach which uses the smoothness equation 3.8 to detect shocks. In both cases, convergence is below one due to the presence of discontinuities. When polynomial degrees of four and above are used, the convergence is relatively unchanged due to limitations in accuracy caused by the time discretization method described by equation 2.31. For both test scenarios, a different CFL number was used depending on the polynomial degree of the solutions. The CFL number was kept constant for a given polynomial as the mesh was refined. In this case, because shocks are present,

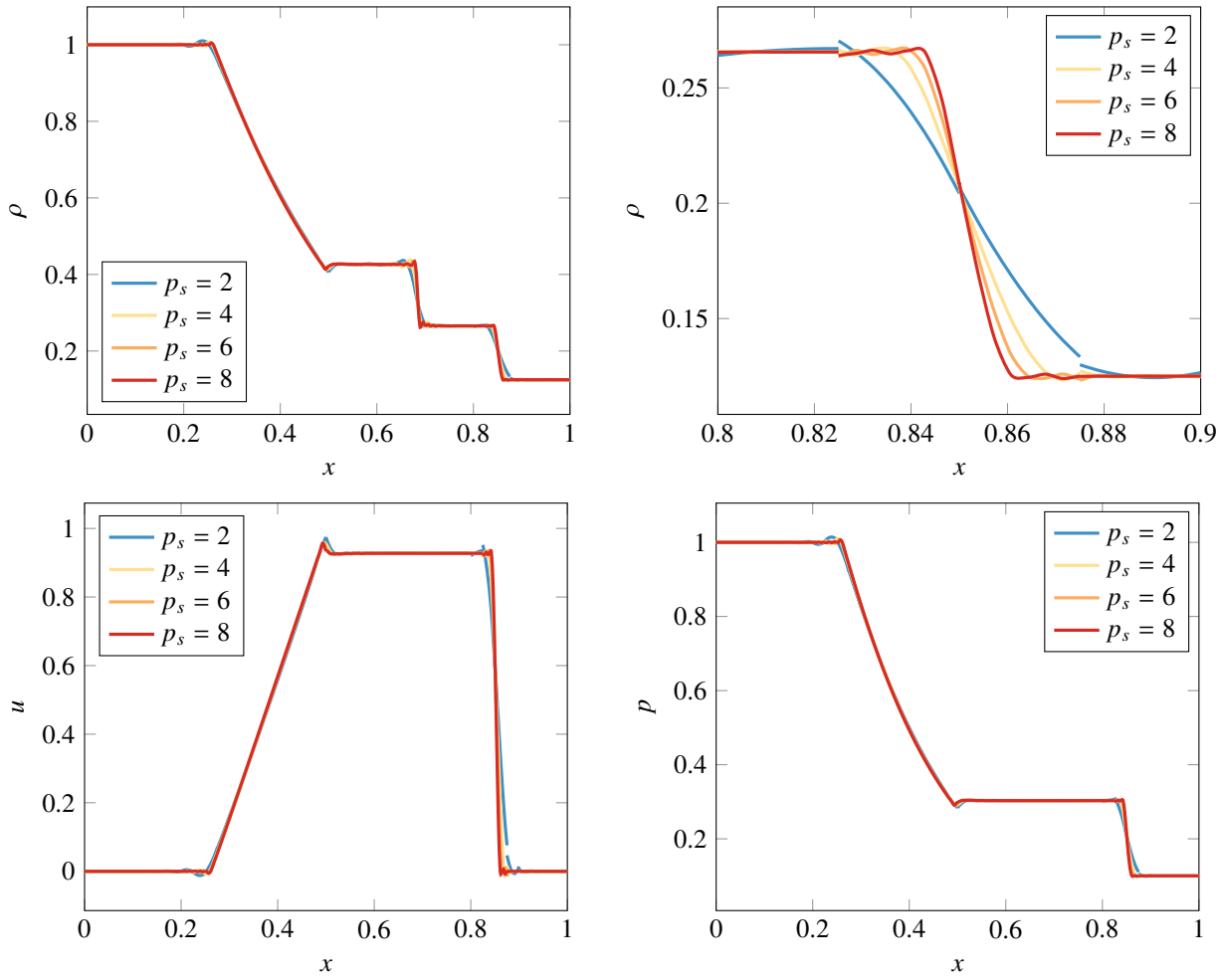


Figure 4.6. Plots of density, velocity, and pressure for Sod's shock tube at $t = 0.2$ using uniform solution polynomials of degree $p_s = 2, 4, 6$ and 8 on a 40 element mesh.

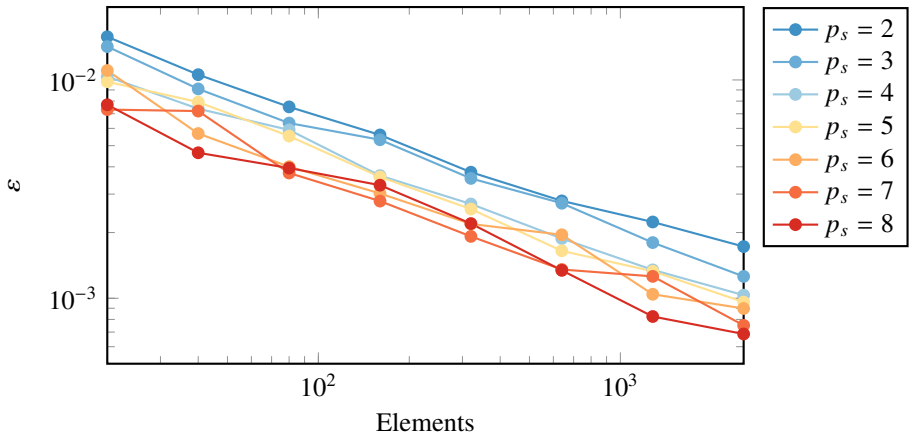


Figure 4.7. Convergence plot of Sod's shock tube using adaptive time stepping with modified dilation shock capturing turned on.

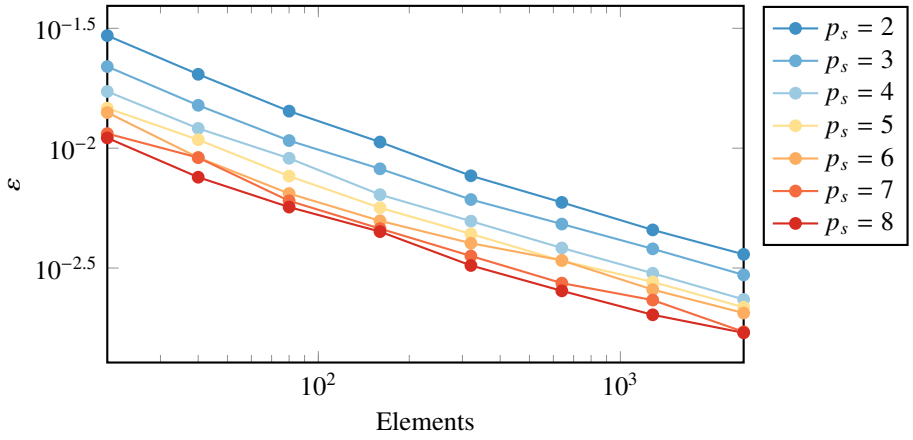


Figure 4.8. Convergence plot of Sod's shock tube using adaptive time stepping with smoothness shock capturing turned on.

CFL values for higher order polynomials needed to be lower than the ones used for the traveling density wave which does not contain any shocks.

4.2 One-Dimensional Cases

In addition to one-dimensional convergence tests for test cases with exact analytical solutions, comparison testing on more complex problems, with no analytical solutions were compared. Both the Shu-Osher and Woodward and Colella blast polynomial increase were performed on a fixed number of elements as well as grid refinement was performed on the highest order polynomial tested. For all three test cases the modified shock capturing and artificial viscosity scheme were utilized. These scenarios were selected because of their prevalence in hypersonic literature, as well as for the interesting interactions that occur in these test cases [12, 17, 39, 40, 41].

4.2.1 Shu-Osher

The initial conditions for the Shu-Osher test case are described in equation 4.3, which splits the domain in two parts. The computational domain is $x \in [0, 1]$ with the left hand side covering $0 \leq x \leq 1/8$, while the right hand side covers $1/8 < x \leq 1$. The final time is $t = 0.178$. The Shu-Osher test case simulates a normal shock moving through a sinusoidal perturbed density field. Although such a scenario is not possible in the real world, testing such interactions provides valuable insight on a solver's ability to resolve non-linear interactions between shocks and smooth distributions. As the shock front travels through the sinusoidal density distribution oscillations accumulate behind it, causing a sharp but smooth oscillatory structure in its wake. Further in the wake of the shock, remnants of the sinusoidal density distribution are found which have been affected by the traveling shock front.

$$\begin{pmatrix} \rho_L \\ p_L \\ u_L \end{pmatrix} = \begin{pmatrix} 3.857143 \\ 10.3333 \\ 2.629369 \end{pmatrix}, \quad \begin{pmatrix} \rho_R \\ p_R \\ u_R \end{pmatrix} = \begin{pmatrix} 1 + 0.2\sin(8x) \\ 1.0 \\ 0.0 \end{pmatrix}, \tag{4.3}$$

Solving the Euler conservation equations, figure 4.9 shows the results for polynomial degrees varying between 2 and 8, and a mesh composed of 96 elements. It can be seen that as the polynomial increases, the shock is resolved more sharply. In addition, the oscillations behind the shock are sharper as the value of the polynomial is increased. Visually, there appears to be little difference between the 6^{th} and 8^{th} order polynomial.

It can be observed in figure 4.9, that as the polynomial degree within the element is increased, the method is capable of a better representation of the solution. Although the maximum polynomial of $p_s = 8$ appears to give the best shock resolution as well as the best resolution of the oscillations behind the shock, $p_s = 6$ is a close second. In light of the increased computational cost and the time stepping accuracy limits, a case could be argued for opting for a lower polynomial degree while still maintaining a high level of accuracy in the results.

Figure 4.10 compares varying levels of mesh refinement for $p_s = 8$. The maximum number of elements tested was 768 which can be considered to be the exact solution for comparison purposes. It appears that for 96 and 192 elements, the shock front is resolved with only a few elements and is significantly sharper than the result from the mesh with 48 elements. Similarly to the Sod test case, different CFL numbers were used based on the polynomial degree. Once a CFL value was selected, the same value was used as the mesh was refined. For higher order polynomials, the CFL values used were once again smaller than for the Sod test case. This can be explained by the stronger shocks and more complex interactions between the shocks and the oscillatory density distribution which leads to more artificial viscosity being added in shock regions, inevitably altering the conservation equations being resolved.

Comparison, in figure 4.10, of the solution for varying levels of mesh refinement indicates, as would be expected, that as the number of elements is increased, the solution get more accurate. It should be noted that in the region $0.2 \leq x \leq 0.5$,

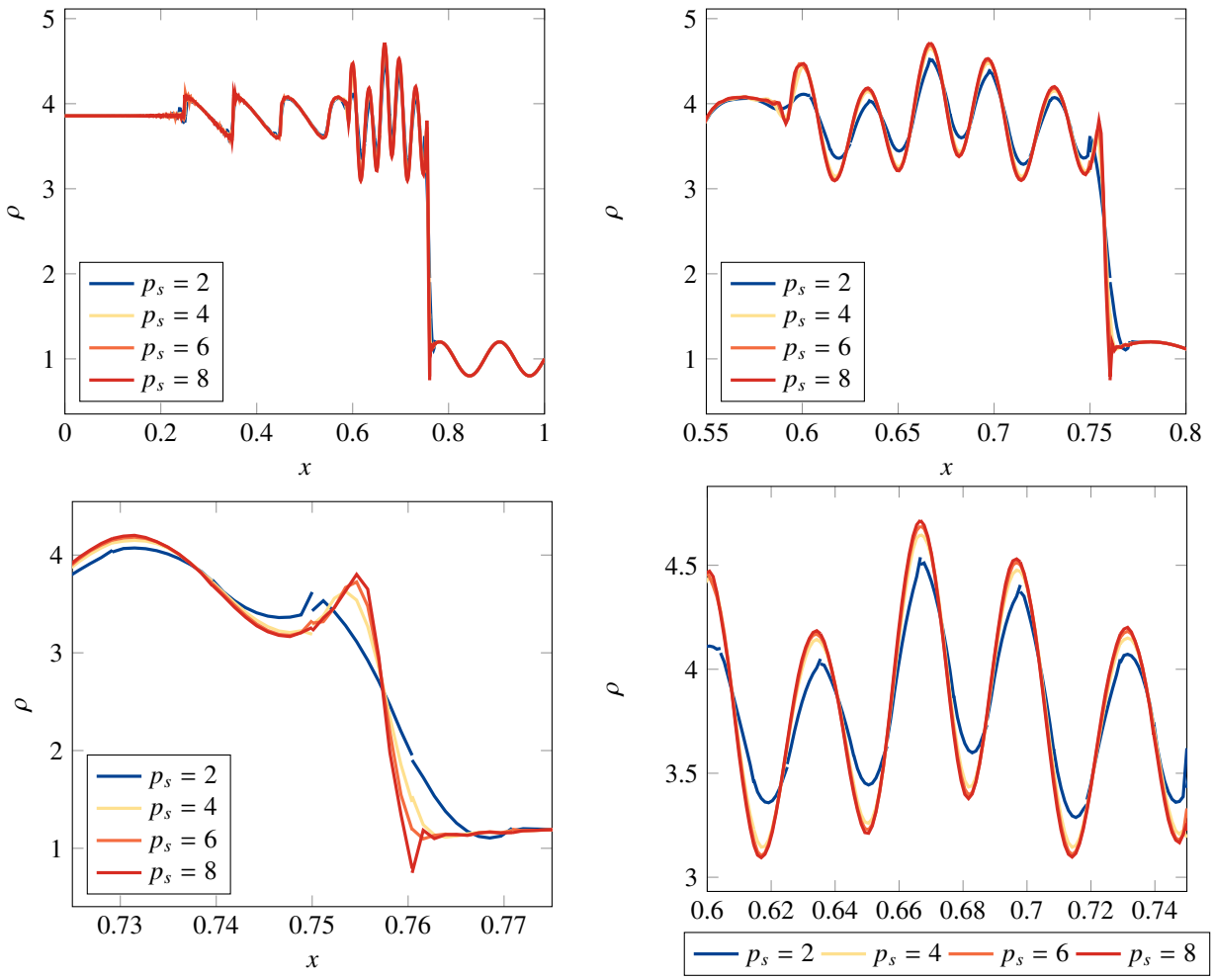


Figure 4.9. Shu-Osher problem with polynomial refinement for a 96 element mesh.

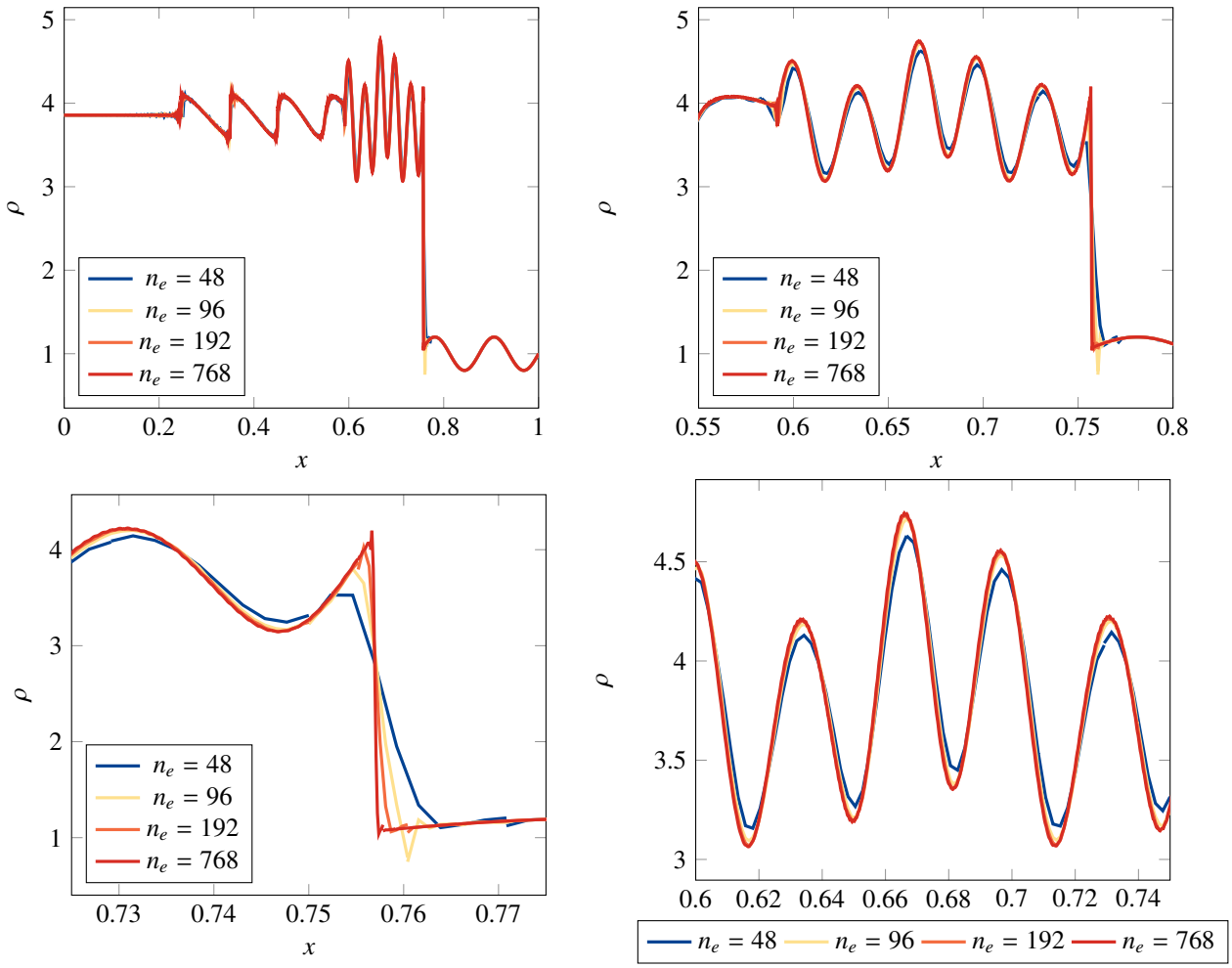


Figure 4.10. Shu-Osher problem with grid refinement for polynomial 8 with maximum mesh size of 768 elements.

all levels of mesh refinement have oscillations present in the solution. Additionally, it can be observed, comparing a mesh with $n_e = 768$ and one with $n_e = 192$, there is very little benefit to quadrupling the number of elements other than a slightly better resolution of the shock. It is also noteworthy that for $n_e = 96$ there is very little difference with $n_e = 768$ in the oscillatory region behind the shock for $0.6 \leq x \leq 0.75$.

4.2.2 Woodward and Colella Blast Wave

The initial conditions for the Woodward and Colella blast wave problem are described using equation 4.4. These conditions split $x \in [0, 1]$ into three pressure zones: two high pressure zones at the edges of the domain and one low pressure zone at the middle, with a uniform density and zero velocity through the entire domain. The left zone is delimited by $0 \leq x \leq 0.1$ and the right zone is delimited by $0.9 \leq x \leq 1.0$. The remainder represents the center portion. The final solution time for which density plots are compared is $t = 0.038$. This problem involves the collision of two shock waves, providing insight in a solver's ability to resolve such interactions. This is usually a difficult test case to resolve because of the multiple interactions with rarefactions and contact discontinuities. At $t = 0$ the two high pressure regions at the edges drive shocks towards the center of the domain while strong rarefaction waves are created in the direction of the walls. As the problem evolves, the shocks move towards the center and rarefaction waves bounce off the walls and interact in a non-linear manner. A more detailed description of these interactions can be found in [42], along with results from other numerical solvers in [12, 16, 17, 43].

$$\begin{pmatrix} \rho_L \\ p_L \\ u_L \end{pmatrix} = \begin{pmatrix} 1.0 \\ 1000.0 \\ 0.0 \end{pmatrix}, \quad \begin{pmatrix} \rho_C \\ p_C \\ u_C \end{pmatrix} = \begin{pmatrix} 1.0 \\ 0.01 \\ 0.0 \end{pmatrix}, \quad \begin{pmatrix} \rho_R \\ p_R \\ u_R \end{pmatrix} = \begin{pmatrix} 1.0 \\ 100.0 \\ 0.0 \end{pmatrix}, \tag{4.4}$$

The interactions between shocks, discontinuities and rarefaction waves described through the Euler conservation equation makes the Woodward and Colella blast wave problem a very interesting benchmark case for testing the validity of a numerical solver. In figure 4.11, the density distribution for $t = 0.038$ is shown for $p_s = 2, 4, 6, 8$ with a grid composed of 80 elements. The results from the simulations show, as was observed in the Sod and Shu-Osher test cases, that, at the shock, a steeper gradient in the solution is obtained as the polynomial degree is increased. It appears though that for the shock on the left hand side of figure 4.11, depicted on the bottom left, there is little difference between $p_s = 4$, $p_s = 6$ and $p_s = 8$. The only significant difference is with $p_s = 2$ for which the solution of the shock front spans many more elements than the other polynomial degrees. The right moving shock, seen in the bottom right of figure 4.12, appears to have a more significant correlation between number of elements required to resolve the shock and polynomial degree. However, for $p_s = 6$ and $p_s = 8$ the steepness of the solution appears to only differ very slightly.

The comparison of various levels of grid refinement for the same polynomial degree, $p_s = 8$, is depicted in figure 4.12.

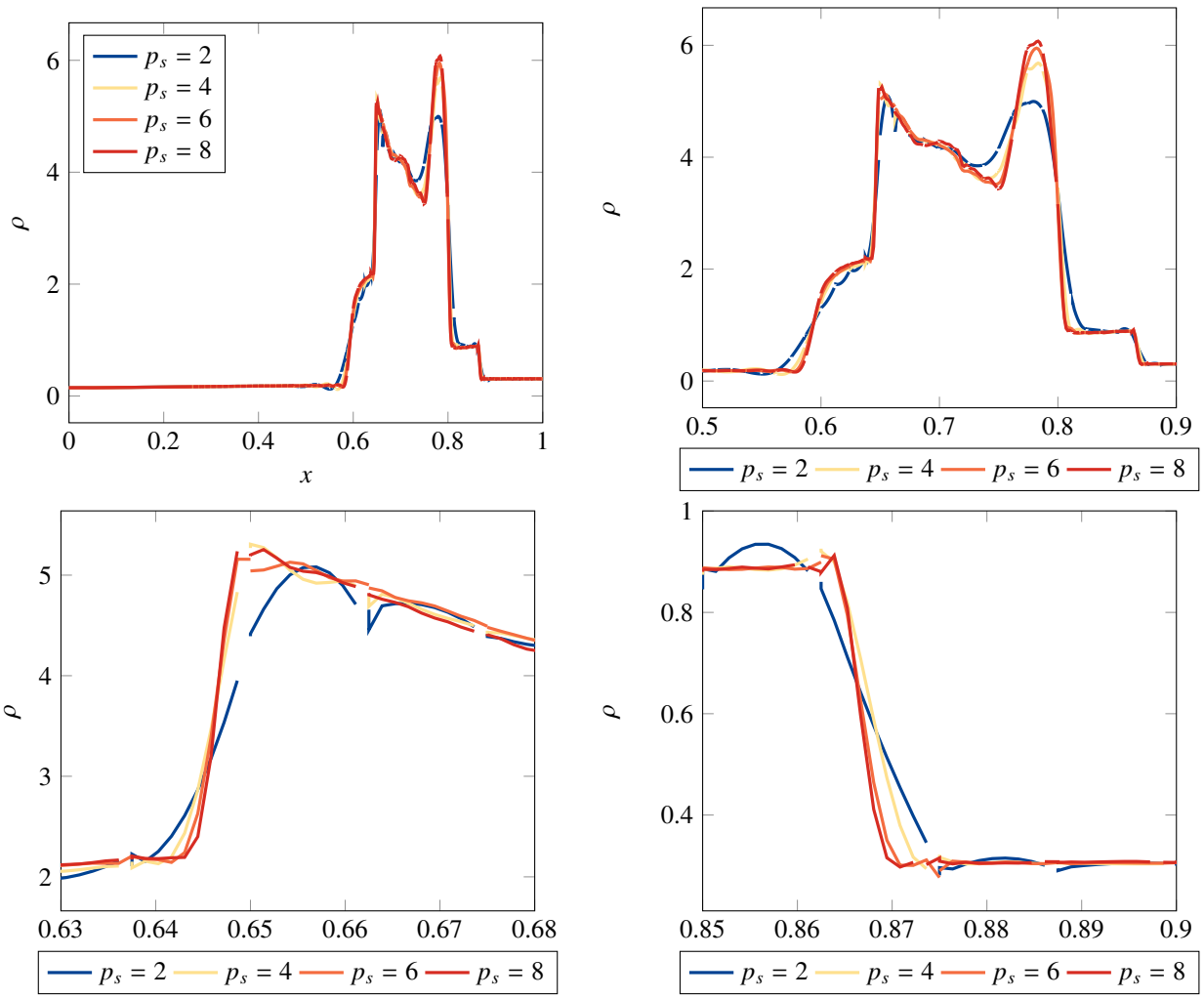


Figure 4.11. Woodward and Colella Blast with polynomial refinementfor 80 element mesh.

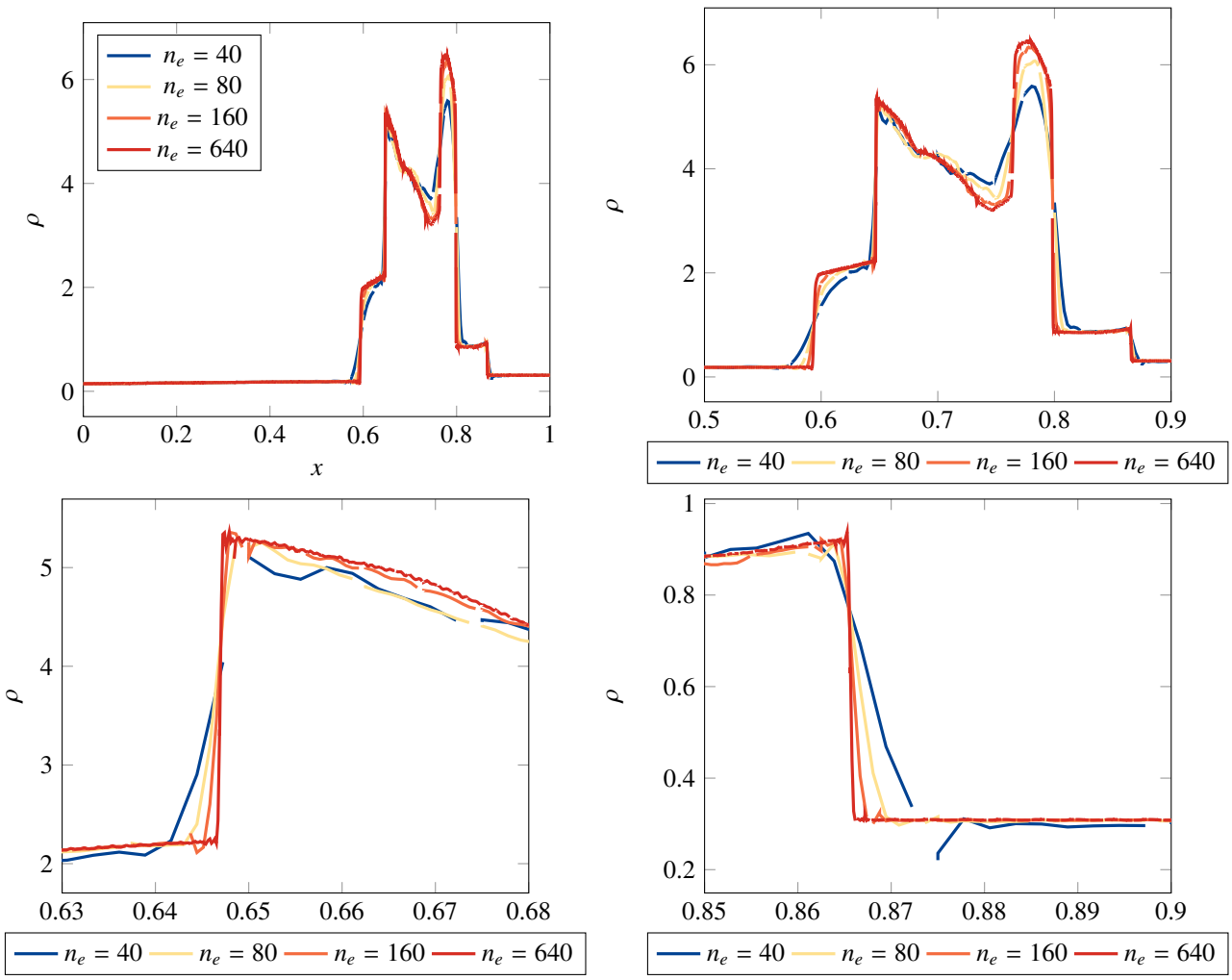


Figure 4.12. Woodward and Colellas Blast with grid refinement for polynomial 8 with maximum mesh of 640 elements.

As expected, the maximum level of refinement, $n_e = 640$ offers the sharpest solution for the shocks moving to the left and right, which are depicted at the bottom of figure 4.12. A similar scenario as the one seen in figure 4.10 for the Shu-Osher case is that the difference between the solution for $n_e = 160$ and $n_e = 640$ is very small, which would indicate that the computational requirements of solving a mesh that is four times more refined outweighs the accuracy gained. In the same manner as with the one-dimensional test cases, a different CFL number was used for different polynomial degree. For a given polynomial degree, the same CFL value was used regardless of the number of elements present in the domain. The shocks in the Woodward and Collela blast are significantly stronger than those of the Sod shock tube or the Shu-Osher test cases and, for this reason, the use of the positive limiter routine was required. Although it was not used at every time step, without it turned on, the occasional presence of negative pressures causes the solver to crash.

For the three test cases in which shocks are presents; Sod, Shu-Osher, and Woodward and Colella blast, a similar trend is observed. As the polynomial degree is increased for a mesh with the same number of elements, shocks are resolved more steeply and the overall solution is more representative of what is expected analytically or when compared to other types of numerical solvers. There also appears to be a limit at which additional increase in the polynomial degree does not yield a significant increase in the level of accuracy, most likely because of the limitations of the temporal scheme. Additionally, as the grid is refined and the number of elements increases for the same polynomial degree, the solution becomes more accurate and gradients become steeper. Here, too, a limit at which the computational cost of further refinement outweighs the accuracy benefits gained.

4.3 Convergence of Two-Dimensional Exact Case

The isentropic vortex is a self sustained vortex which theoretically should run through an inviscid domain with periodic boundary conditions without dissipation. This idealized scenario means that the solution for a vortex with initial conditions given by equation 4.5 can be solved analytically at any given time.

$$\begin{aligned} \rho &= \left(1 - \frac{S^2 M^2 (\gamma - 1) e^{2f}}{8\pi^2}\right)^{1/(\gamma-1)} \\ u &= \frac{S(y - y_C) e^f}{2\pi \mathrm{R}} \\ v &= \frac{S(x - x_C) e^f}{2\pi \mathrm{R}} \\ p &= \frac{\rho^\gamma}{\gamma M^2} \\ f &= \frac{1 - (x - x_c)^2 - (y - y_c)^2}{2\mathrm{R}^2} \end{aligned} \tag{4.5}$$

In the initial conditions functions described in equation 4.5, S determines the strength of the vortex, R determines the radius of the vortex, M is the Mach number of the vortex which will determine its speed, x_c and y_c are the x and

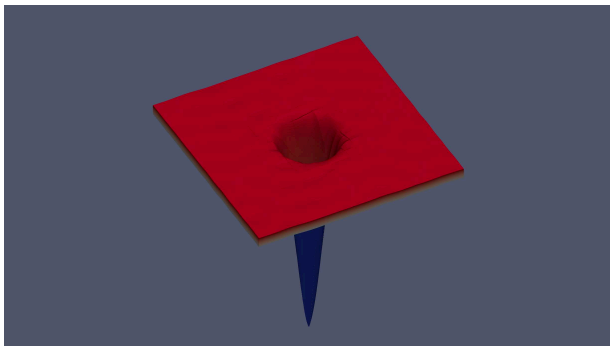


Figure 4.13. Density distribution of the isentropic vortex after one pass through the domain.

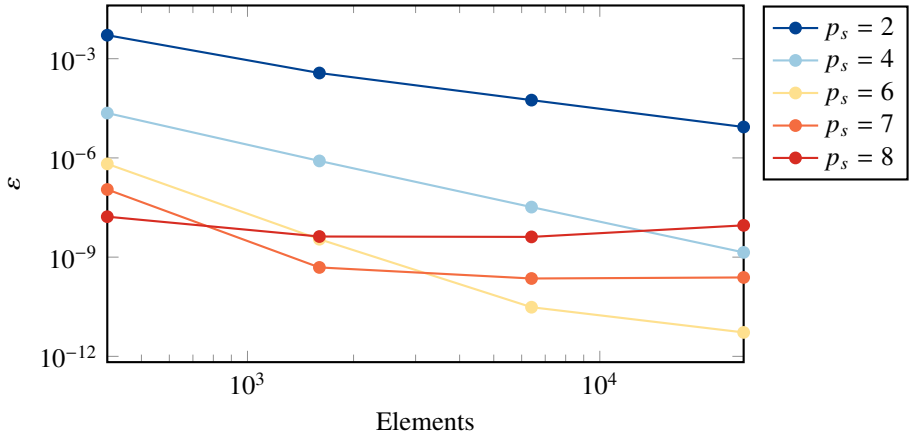


Figure 4.14. Convergence plot of isentropic vortex problem using adaptive time stepping and no shock capturing.

y coordinates for the center of the vortex, and γ is the heat capacity ratio of the fluid, typically $\gamma = 1.4$ for air. The conservation equations being solved in all polynomial cases are the inviscid Euler equations, meaning the vortex should theoretically run indefinitely.

An additional area to be investigated is that the presence of the shock capturing and artificial viscosity function do not add unwanted viscosity to the solution, which would break down the fluid structure as the vortex moves through the domain. In order to determine if this is the case, convergence of the solver is tested and compared for varying polynomial degrees with the modified shock capturing turned on and off and using adaptive time stepping. For all polynomial degrees, a different CFL number was used and maintained for a given polynomial as the mesh was refined.

In Figure 4.14 and 4.15 the convergence for $p_s = 2, 4, 6, 7, 8$ is plotted for increasing mesh sizes of 400, 1600, 6400 and 25600 elements. In all cases, the convergence with shock capturing turned on and off is almost identical, indicating that the shock capturing method does not add unwanted viscosity in regions where shocks are not present. It is noteworthy to point out that the convergence rate for $p_s = 8$ doesn't follow the increasing trend in convergence as the polynomial degree is increased. Figures 4.14 and 4.15 also indicate that for $p_s = 7$ and $p_s = 8$, increasing the number of elements does not increase accuracy because of unwanted oscillations in the solution polynomial within a given element.

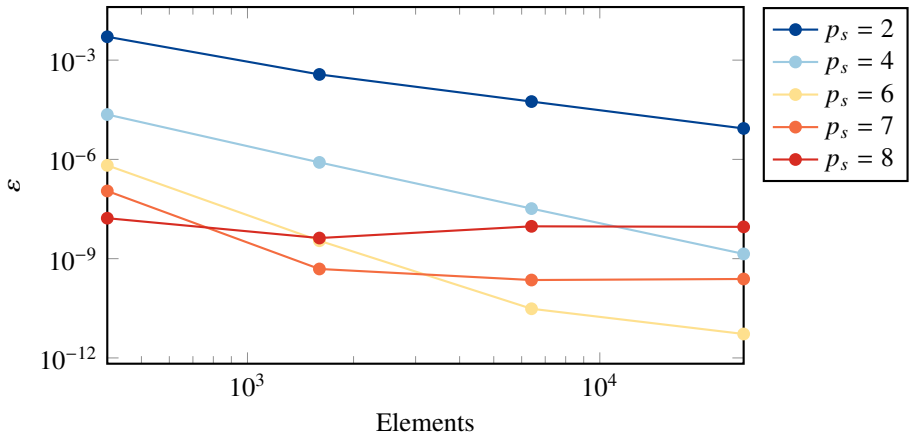


Figure 4.15. Convergence plot of Isentropic Vortex using adaptive time stepping with dilation shock capturing turned on.

Additionally it appears that accuracy above 4th order is not obtained, likely because of the time stepping scheme which is 4th order accurate. Table 4.2 gives the convergence order for varying polynomial degrees, comparing the results for shock capturing on and off.

Table 4.2. Rate of convergence for Isentropic Vortex with Dilation routine On and Off.

p_s	$DilationOff$	$DilationOn$
2	1.8984	1.8913
4	2.4032	2.4032
6	3.7830	3.7830
7	3.9159	3.9159
8	0.9963	0.9963

4.4 Convergence of Complex Two-Dimensional Cases

Similarly to the 1D testing validation, test cases without an exact solution are also explored in two-dimensions in order to compare results using the dilation based shock capturing technique described in equation 3.16 along with the artificial viscosity function 3.20, with previously published works. Although the benchmark cases explored do not have exact solutions, they have been extensively studied by many authors who are in agreement with the expected results that should be obtained. By testing 2D benchmark cases with no exact solutions, more complex interactions between shocks, reflections, rarefaction waves and turbulence can be investigated, and results can be compared to further determine the robustness of the solver and its capabilities.

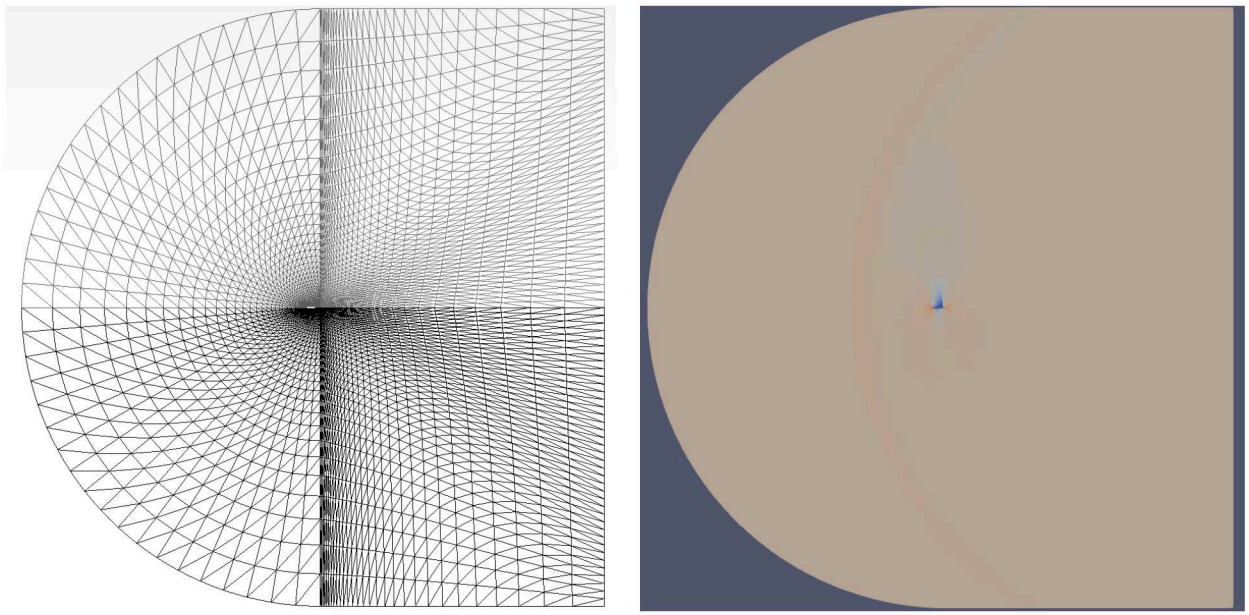


Figure 4.16. NACA 0012 mesh compared to steady state solution for incoming mach number of 0.8 and angle of attack of 1.5°.

4.4.1 NACA 0012 Airfoil

The NACA 0012 airfoil is a symmetrical airfoil with not camber, and has a 12% thickness to chord length ratio. This simple airfoil design was tested in the transonic and supersonic region in order to further determine the solvers ability to resolve complex interactions, this time in multiple dimensions. In both cases, the initial conditions are a uniform density, pressure and velocity distribution described as

$$\begin{aligned} \rho &= 1.0, \\ u &= M\sqrt{\gamma}\cos(aoa), \\ v &= M\sqrt{\gamma}\sin(aoa), \\ p &= 1.0, \end{aligned} \tag{4.6}$$

where M is the incoming Mach number and aoa is the angle of attack. The mesh, shown in figure 4.16(left), consists of 12152 2^{nd} order elements. A representative density field is shown in figure 4.16(right). The first test case for which steady state results are presented in figures 4.17 and 4.18, has $M = 0.8$ and $aoa = 1.5$. This results in a flow over the airfoil in the transonic region. The second case, presented in figures 4.19 and 4.20, has $M = 1.5$ and $aoa = 0.0$, resulting in a supersonic flow over the airfoil.

In figures 4.17- 4.20, results are presented for $p_s = 2$ (left) and $p_s = 8$ (right). For both the transonic and supersonic cases, the density field (figs 4.17 and 4.19) is shown as well as the field of artificial viscosity (figs 4.18 and 4.20). For the transonic case, the flow accelerate to supersonic speeds on the top of the airfoil. A nearly vertical shock is present

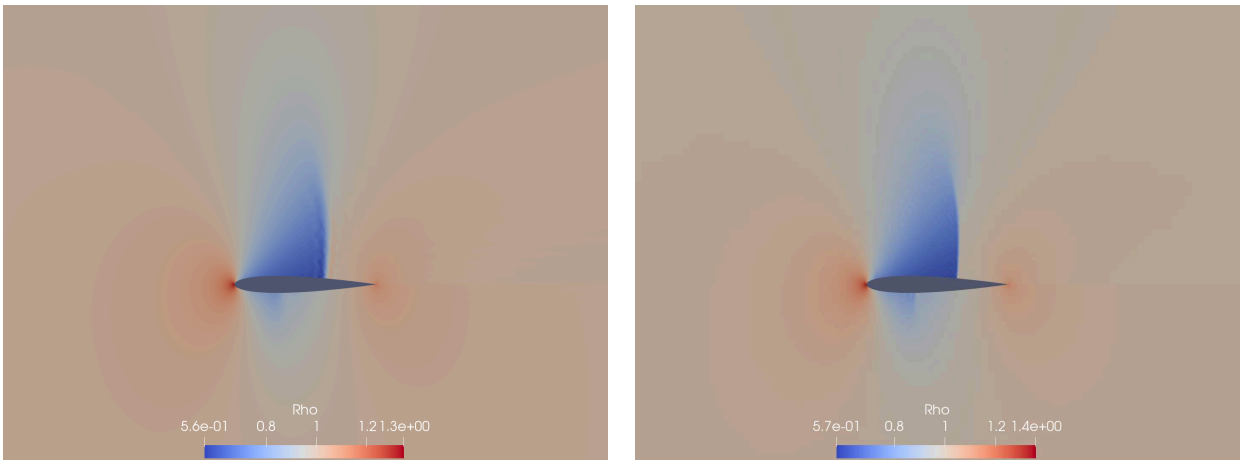


Figure 4.17. Naca 0012 airfoil with angle of attach of 1.5 degrees and incoming flow at Mach 0.8. Comparison of density for polynomial 2 and 6.

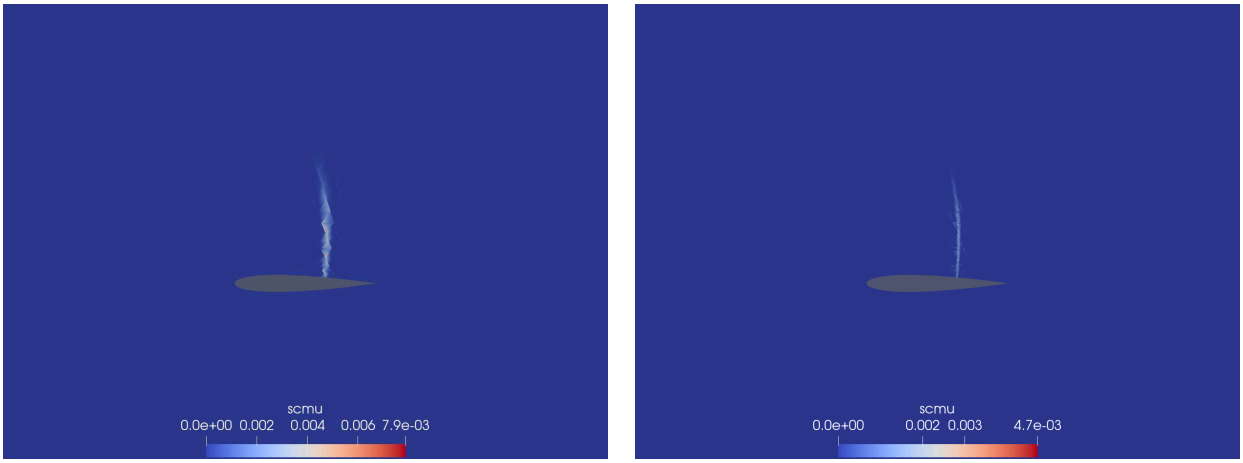


Figure 4.18. Naca 0012 airfoil with angle of attach of 1.5 degrees and incoming flow at Mach 0.8. Comparison of artificial viscosity for polynomial 2 and 6.

above the top, downstream half the airfoil. For $p_s = 2$, the shock is visibly thicker, encompassing more computational cells. For $p_s = 6$, the shock surface is smoother. The region where the artificial viscosity is added is centered exclusively around the added shock. A greater amount of viscosity is added for $p_s = 2$. The "raggedness" and increased thickness of the shock surface is also more evident in figure 4.18. In [40] a similar shrinking of the region where artificial viscosity is added can be found as the mesh is refined. This would suggest that increasing the polynomial degree could have a similar effect to refining the mesh.

Figures 4.19 and 4.20 show the results for the Mach 1.5 supersonic flow on the same mesh containing 12152 2nd order elements for $p_s = 2$ and $p_s = 6$. In both cases there is are clear symmetrical oblique shocks at the nose and tail of the airfoil. The shock at the tip of the airfoil for $p_s = 2$ seen in on the left of Figure 4.19 have a smeared appearance with a rippling effect around it similar to what is seen in Figure 4.17. On the right hand side of Figure 4.19 are the results for the density distribution of $p_s = 6$ for which the shocks at the tip of the airfoil are much clearer with almost no apparent

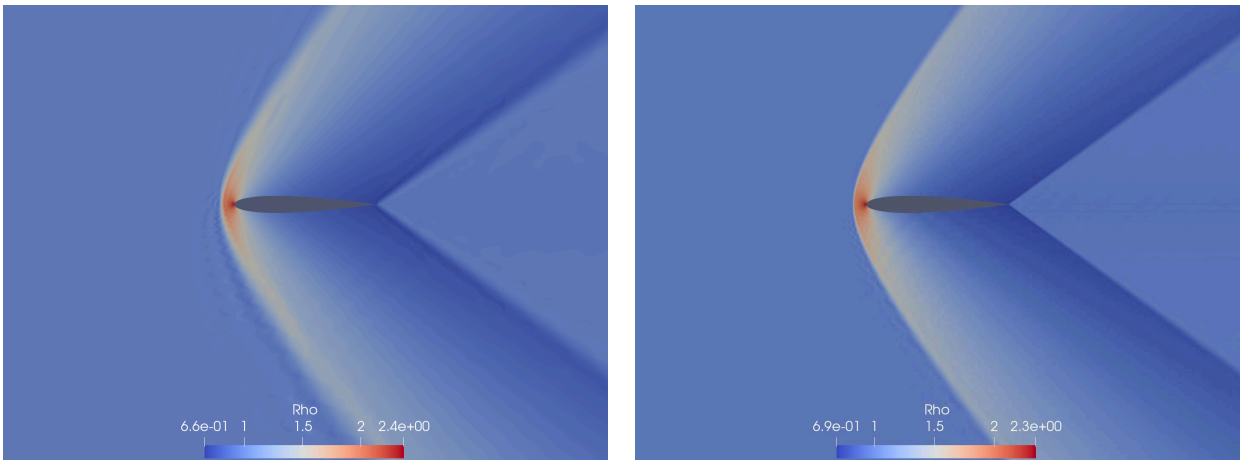


Figure 4.19. Naca 0012 airfoil with angle of attach of 0 degrees and incoming flow at Mach 1.5. Comparison of density for polynomial 2 and 6.

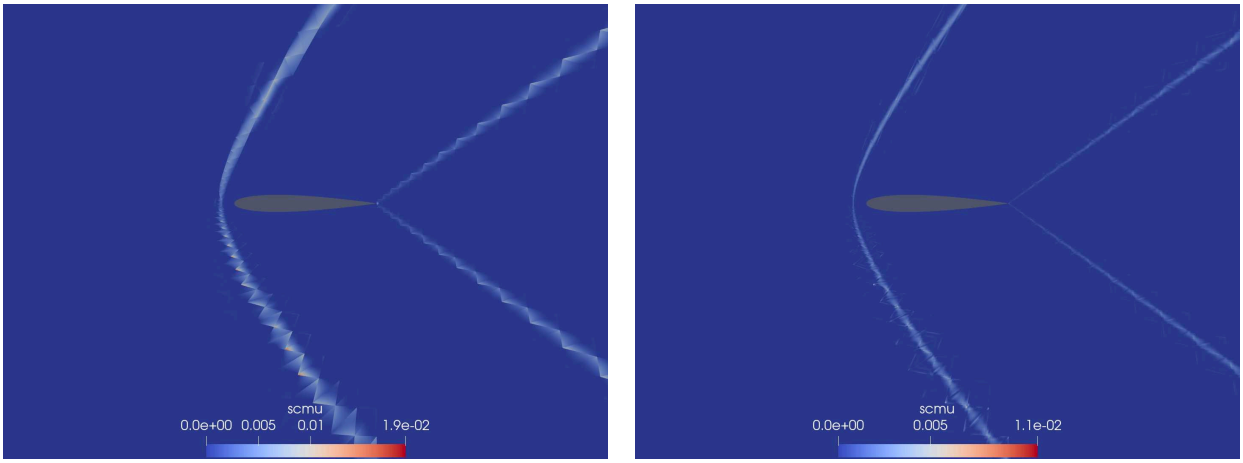


Figure 4.20. Naca 0012 airfoil with angle of attach of 0 degrees and incoming flow at Mach 1.5. Comparison of artificial viscosity for polynomial 2 and 6.

smearing of the shock.

For the supersonic case, symmetrical oblique shocks are located at the leading and trailing edges of the airfoil. Similarly to the transonic case, the leading and trailing shocks are smeared and "rippled" when $p_s = 2$ is used. For $p_s = 6$, the shock surfaces are visibly thinner and smoother. the distributionof artificial viscosity, show in figure 4.20, shows again the shock detector is adding viscosity solely in the shock regions. In order to determine the effectiveness of their shock capturing approach, [14] conducted similar numerical simulations on the NACA 0012 airfoil for both the transonic and supersonic regimes and obtained results similar to the ones presented using a modified shock capturing approach. In both test cases performed in this work the increase in polynomial degree lead to a better resolution of the shocks present in the flow. It is also noteworthy that for the solutions using $p_s = 6$, the regions of artificial viscosity are smaller, which would indicate that the solution of the conservation equations is artificially affected in a smaller proportion of the overall domain, leading to a more accurate overall solution. Similar to results obtained for the

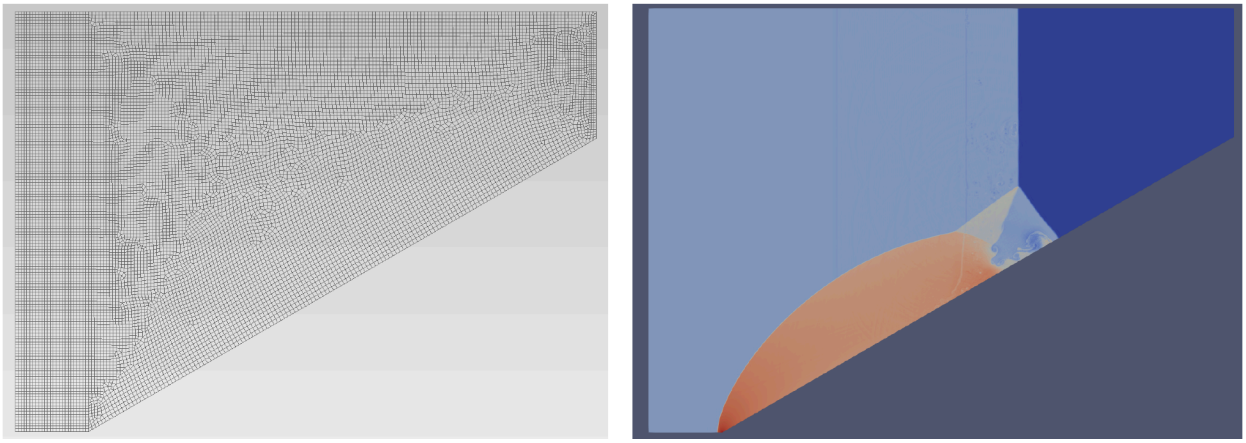


Figure 4.21. Double Mach reflection mesh compared to typical solution at $t = 0.2s$.

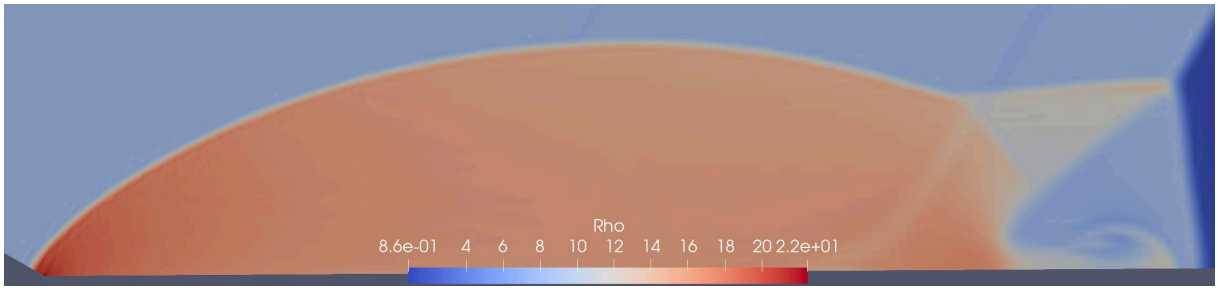
one-dimensional test cases, increasing the polynomial degree could have an effect similar to mesh refinement.

4.4.2 Double Mach Reflection

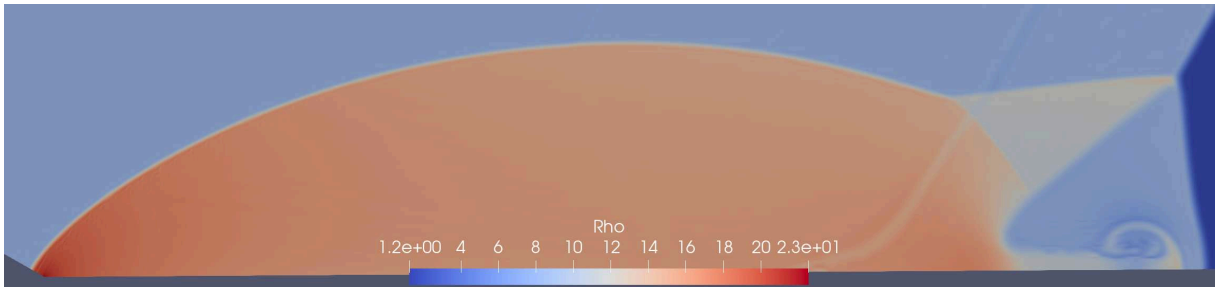
A double Mach reflection was simulated on an unstructured mesh containing 15063 cells, as seen in figure 4.21 for $p_s = 2$, $p_s = 4$, $p_s = 6$ and $p_s = 8$. Results are presented and compared at $t = 0.2s$ in Figures 4.22, 4.23, 4.24 and 4.25. The initial conditions for the double Mach reflection are a uniform density $rho = 1.4$, pressure $p = 1.0$ and zero velocity. The inclined surface forms an angle of 30° with the horizontal and the incoming flow is at an aoa of 0° and has properties of density $\rho = 8$, pressure $p = 116.5$ and velocity 8.25 which gives a Mach number $M = 10$. This test case has a strong shock present traveling through the domain at Mach 10 along with complex re-circulation areas behind the primary Mach stem. The investigation into the solver's ability to capture inviscid Euler equations for the double Mach reflection further solidifies the ability for HORUS to resolve strong shocks with complex interactions.

Figure 4.22 shows the results of the density field at $t = 0.2s$ for increasing polynomial degrees. In all four cases, the different regions, as described by [44], are distinguishable, with some smearing present for lower polynomial degrees and clearer delimitations for $p_s = 6$ and $p_s = 8$. The vortex-like structures along the primary slip line are clearly distinguishable for $p_s = 8$ while for $p_s < 8$ the primary slip line area appears blurry and no visible disturbances are present. When comparing results in figure 4.23, it appears that for $p_s = 8$, numerical artifacts are present behind the incident shock wave which can be explained by the addition of artificial viscosity in this area. For other polynomial degrees, the shock is resolved over a larger number of cells, which might explain why the artifacts aren't as easily visible in the density distribution. In all four test cases, an unphysical wave, as described by [45], trailing behind and parallel to the incident shock is present. This wave is said to be caused by the initial conditions of the flow and is not a phenomenon that would be present in nature. The thickness of this wave is also diminishing as the polynomial degree is increased.

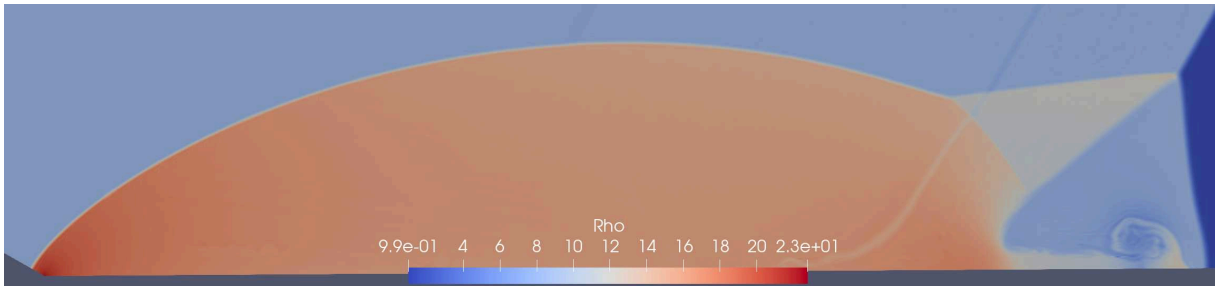
The distribution of artificial viscosity for $p_s = 2$, $p_s = 4$, $p_S = 6$ and $p_s = 8$ are presented in Figures 4.24 and 4.25.



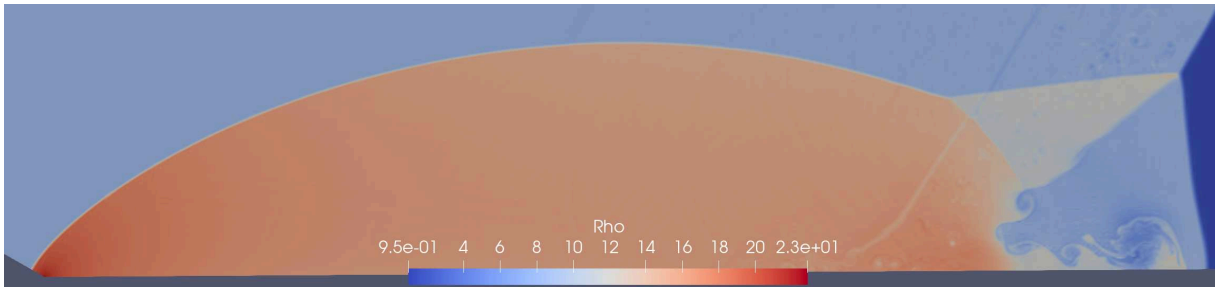
(a) $p_s = 2$



(b) $p_s = 4$

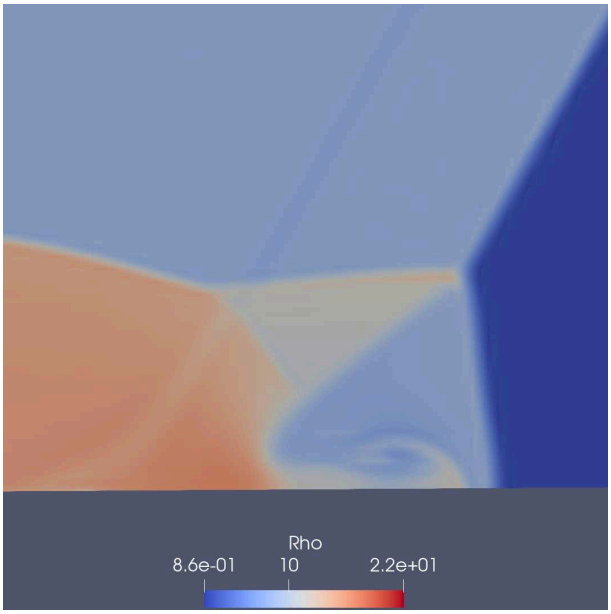


(c) $p_s = 6$

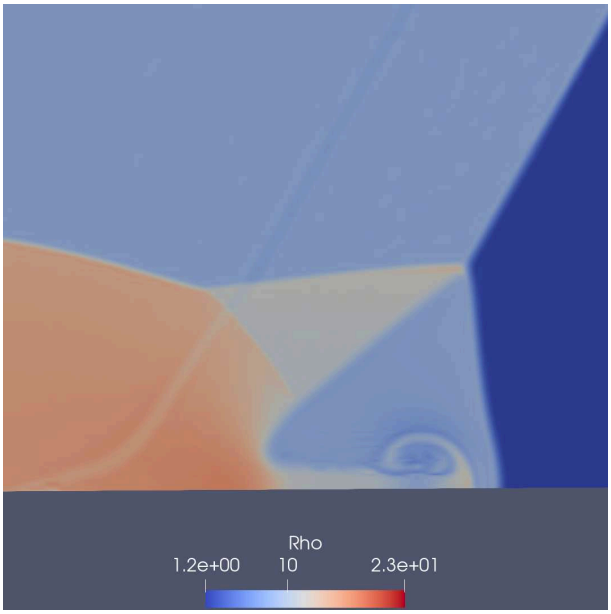


(d) $p_s = 8$

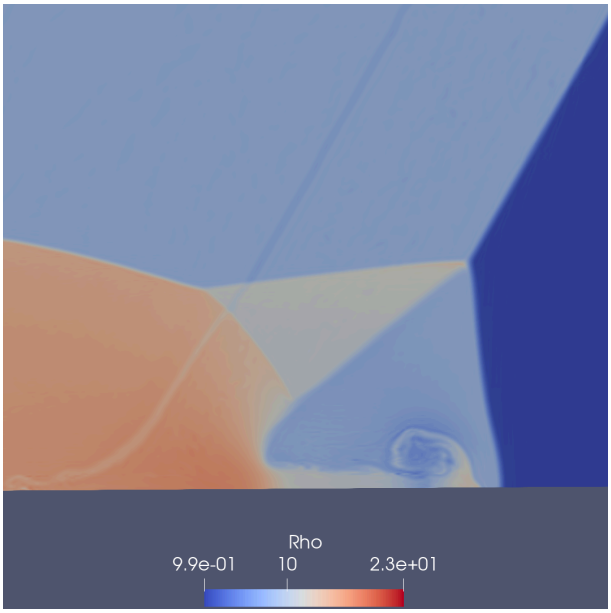
Figure 4.22. Density for double Mach reflection with varying polynomial degrees.



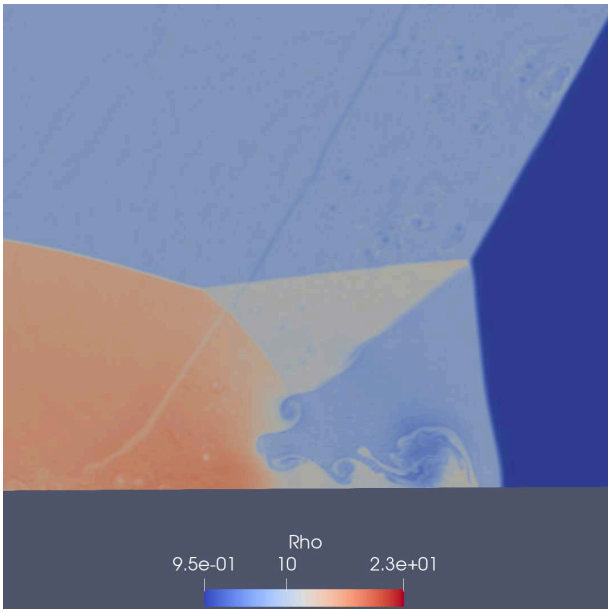
(a) $p_s = 2$



(b) $p_s = 4$

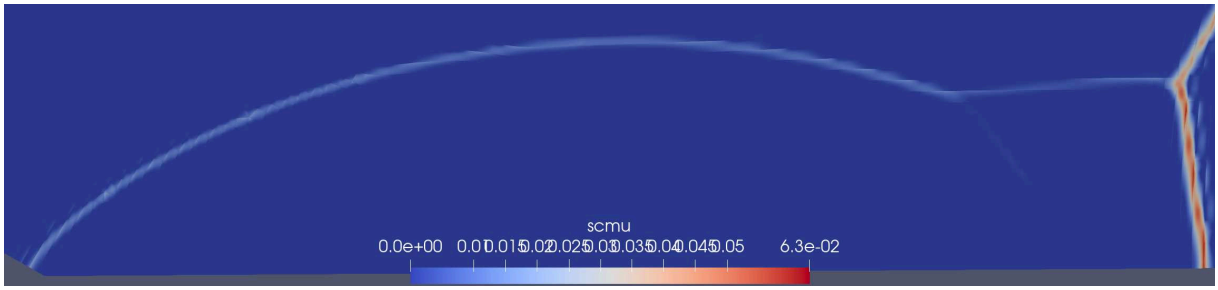


(c) $p_s = 6$

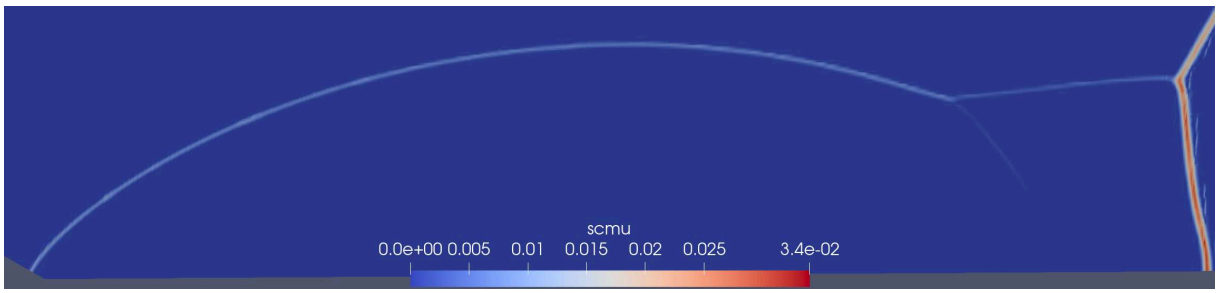


(d) $p_s = 8$

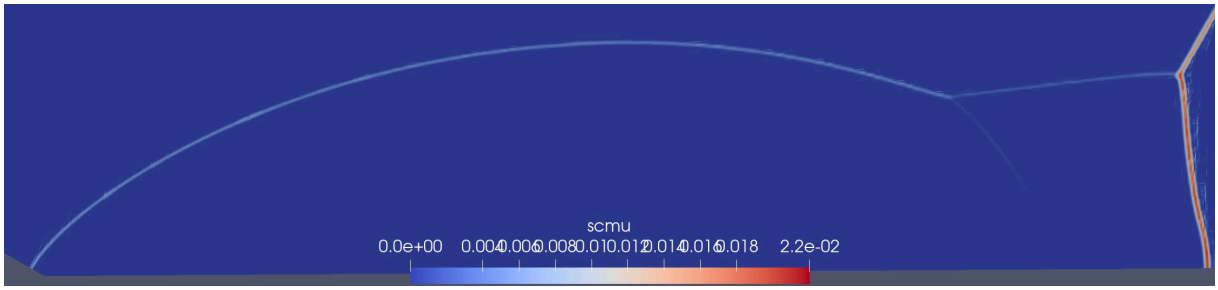
Figure 4.23. Close up of density in the vicinity of the primary slip line and mach stem for the double Mach reflection problem with varying polynomial degrees.



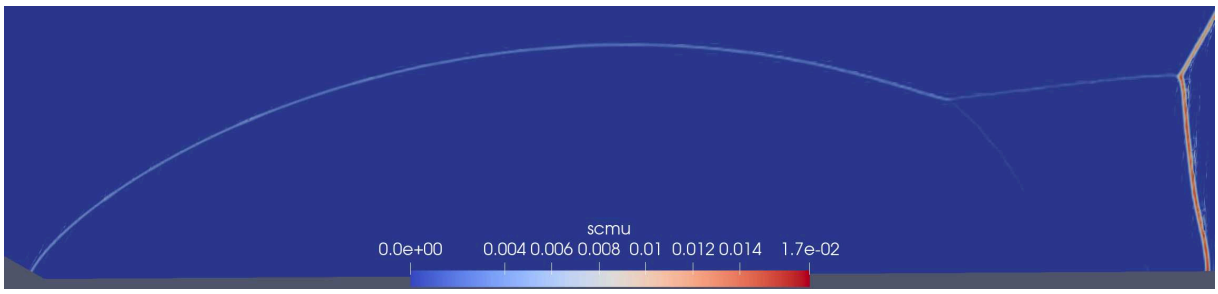
(a) $p_s = 2$



(b) $p_s = 4$



(c) $p_s = 6$



(d) $p_s = 8$

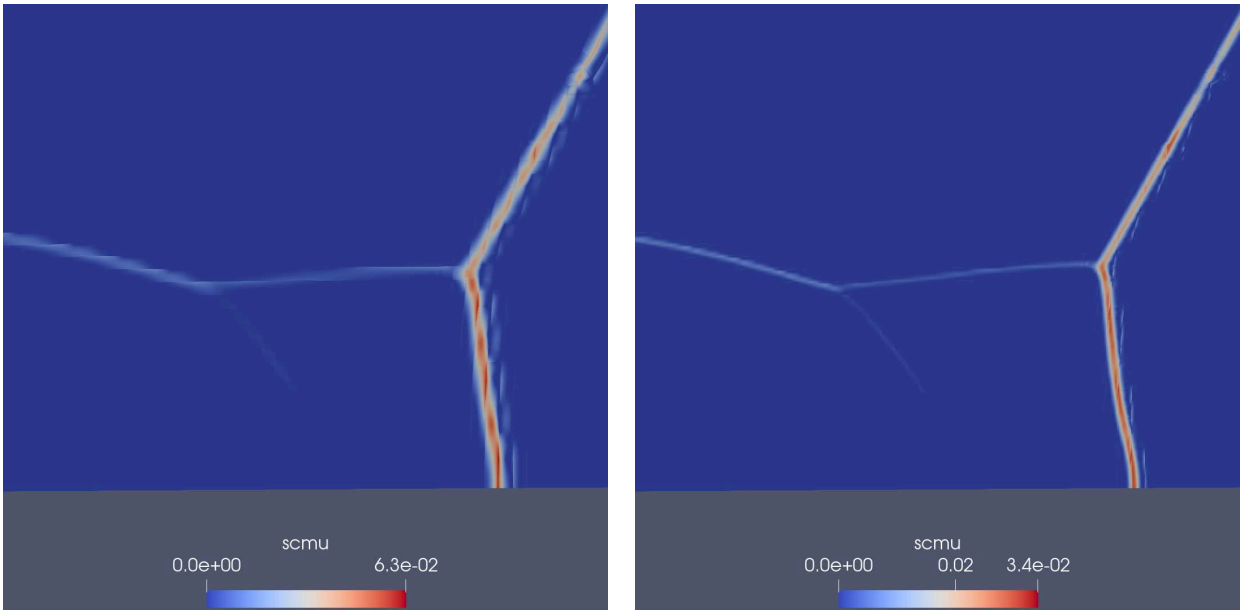
Figure 4.24. Artificial viscosity for the double Mach reflection with varying polynomial degrees.

For all cases, artificial viscosity is added at the incident shock wave, primary Mach stem, primary reflected shock wave, secondary reflected shock wave and secondary Mach stem. In all four cases, affected cells ahead of the incident shock wave and primary Mach reflection are present. As the polynomial degree increases, the number of cells affected decreases. Additionally, as the polynomial degree is increased, the thickness of the region where the majority of the artificial viscosity is added decreases. The primary area affected by this is the incident shock wave and the primary Mach stem. The thickness of the primary reflected shock wave and secondary Mach stem are affected, to a lesser extent, by the increase in polynomial degree as well. In the case of $p_s = 2$, the secondary reflected shock wave covers more cells, in thickness, than all other polynomials degrees, which appear to affect the same number of cells regardless of polynomial degree. Interestingly, when comparing the regions where artificial viscosity is added with the results of [46], it can be observed that the dilation approach does not add viscosity in the re-circulation region, which would be expected since no shock are present.

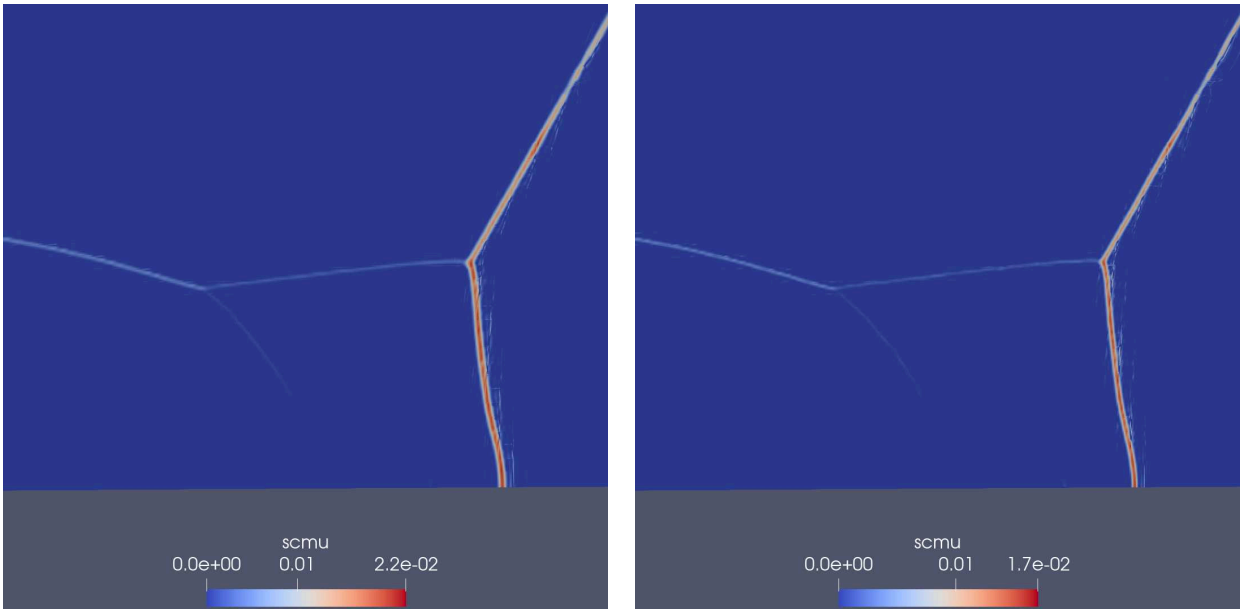
The double Mach reflection is a difficult case for the HORUS solver to run. More specifically, without the use of the positive limiter, cells with negative pressure appear in the domain. Some of these cells are sometimes present in regions in the domain that are not of interest, but on occasion they also affect the area being studied. In no situation does the number of cells being affected increase to cover large regions of the flow, and these areas are never the same as the simulation progresses through time. The positive limiter acts as a safety which stops the pressure from being negative and therefore allows the simulation to keep running without interruption. In all cases, the use of the positive limiter does not appear to affect the solution and results obtained are comparable to numerical results presented in [12, 16, 44, 45, 47]. Although an analytical solution was developed in [48] only numerical results are compared. It is also apparent that as the polynomial degree is increased the thickness of the regions where shocks are present is drastically reduced and significantly less smearing of the solution is present in the domain. For all two dimensional test cases, a correlation between polynomial degree and sharpness of the shock is noted and increasing the polynomial degree appears to yield more accurate solutions in a manner similar to mesh refinement.

4.5 Comparison of Results from HORUS and FV Methods

The results obtained using HORUS were compared with a second order graphic processing unit (GPU) based FV solver. Since the two solvers are running on different hardware types, it is difficult to do a fair comparison of their run-time. The double mach reflection run-time for the GPU solver on a mesh with 50 million elements was approximately 4 hours compared to approximately 36 hours for the DG solver running on 40 CPUs on a mesh of 15063 elements with $p_s = 8$ corresponding to 1220103 DOF. The FV solver is second order accurate in space by using the MUSCL-Hancock scheme. The Riemann solver utilized is HLLC combined with a Von Leer limiter in order to solve single-step Arrhenius Euler equations [49]. This comparison between the two methods is used to determine if the results obtained using the HORUS



(a) $p_s = 2$ (b) $p_s = 4$



(c) $p_s = 6$ (d) $p_s = 8$

Figure 4.25. Close up of artificial viscosity in the vicinity of the primary slip line and mach stem for the double Mach reflection with varying polynomial degrees.

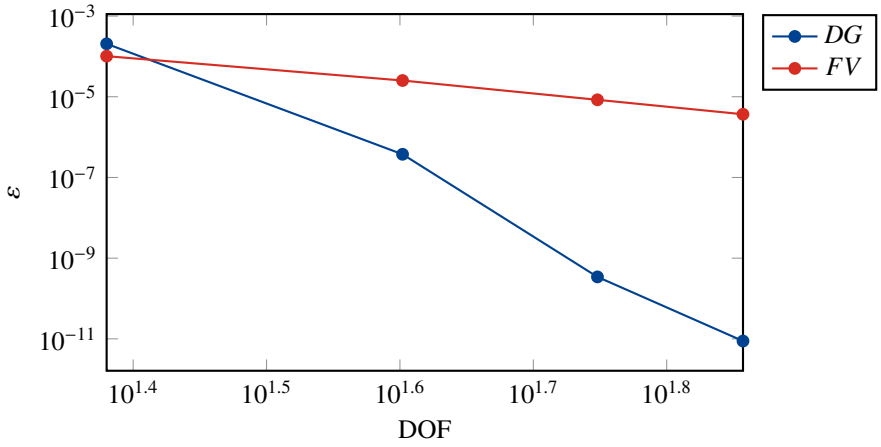


Figure 4.26. Convergence plot of the sine wave density distribution comparing the DG method to the FV method. For varying polynomial degrees on mesh of 8 elements with equivalent DOF using the FV method

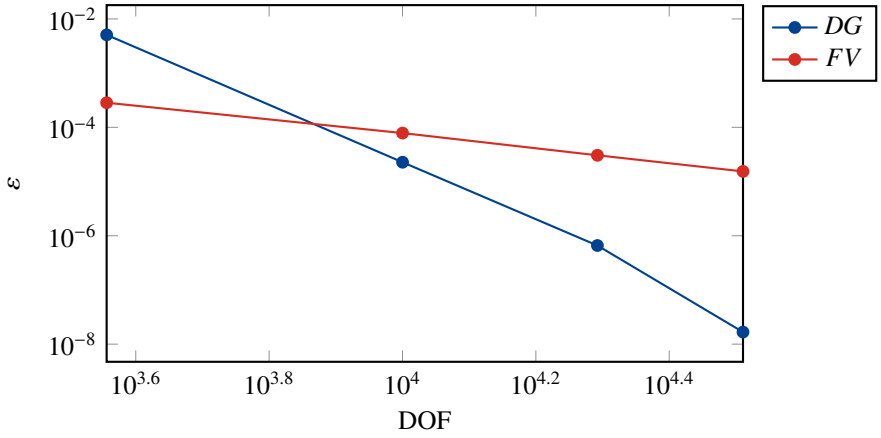


Figure 4.27. Convergence plot of the isentropic vortex comparing the DG method to the FV method. For varying polynomial degrees on mesh of 400 elements with equivalent DOF using the FV method

solver, with its shock capturing routines, is in agreement with other solvers used within the research group. Comparison of the Sod shock tube, Shu-Osher problem, Woodward and Colella blast and the double Mach reflection can be found in figures 4.28, 4.29, 4.30, 4.31 and 4.32 below. For the 1D cases, levels of mesh refinement for the FV method were selected so that the number of degrees of freedom (DOF) have a corresponding polynomial value using the DG method. More specifically, for $p_s = 2$ the coarsest mesh for the FV method has the same number of DOF and subsequently for other values of p_s. For example, in figure 4.26, the the coarsest result in blue represents $p_s = 2$ for the DG method on a 1D mesh of 8 elements, yielding 24 DOF. Consequently, the FV mesh consists of 24 elements or 24 DOF. The other DG points in blue represent $p_s = 4$, $p_s = 6$ and $p_s = 8$ on a mesh of 8 elements.

Figures 4.26 and 4.27 show the $L2$ error for the density wave described in section 4.1, and for the isentropic vortex described in section 4.3. In order to show a fair comparison between the two solvers, the DOF for both methods are the same. In the case of the DG solver, the mesh was fixed and the polynomial degree was varied. On the other hand, for the

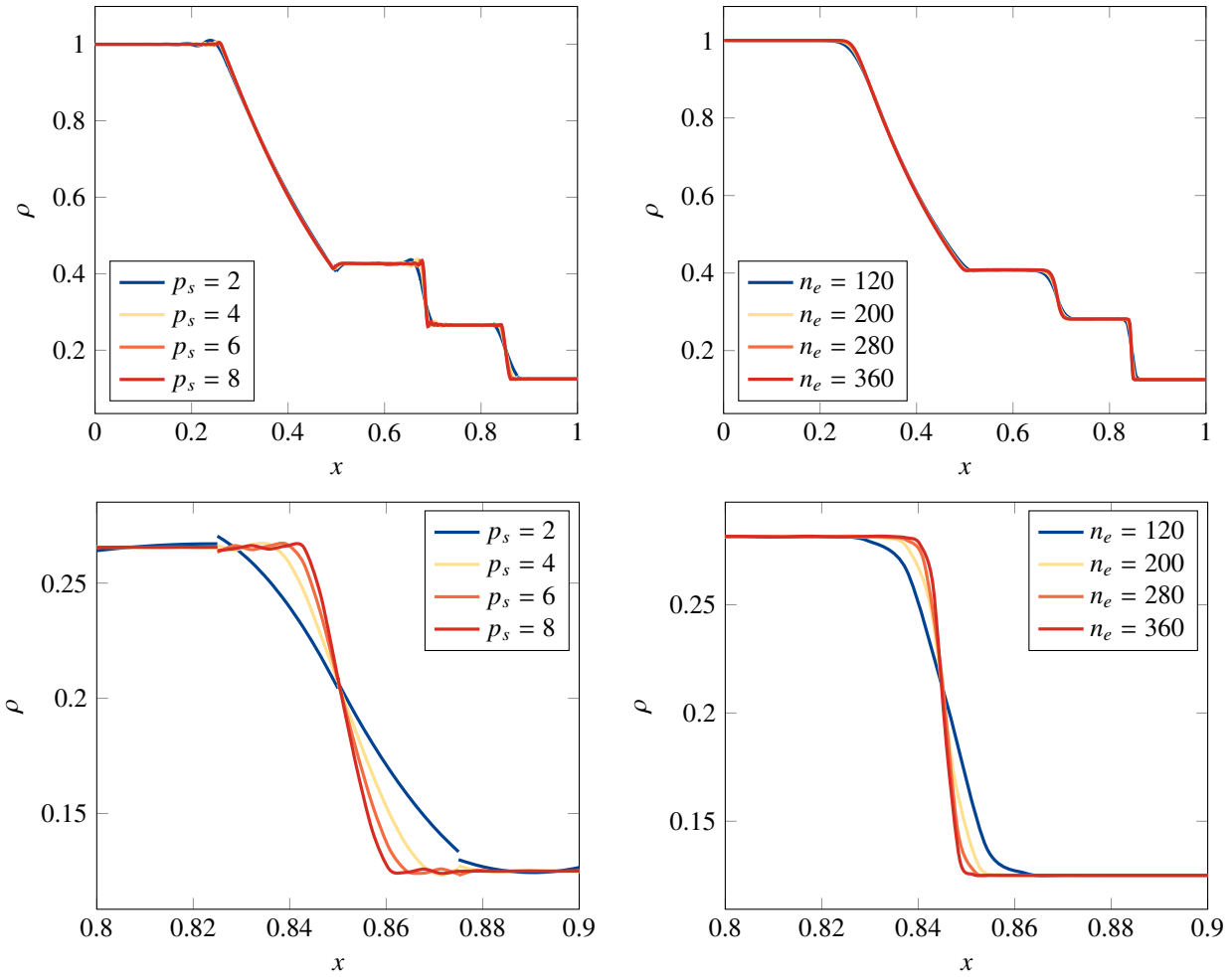


Figure 4.28. Density comparison of the DG method with $p_s = 2$, $p_s = 4$, $p_s = 6$ and $p_s = 8$ on a mesh with $n_e = 40$ (left) against the FV method on meshes with $n_e = 120$, $n_e = 200$, $n_e = 280$ and $n_e = 360$ for the SOD shock tube (right).

FV solver, the mesh was refined in order to match the DOF of the DG method. Figures 4.26 and 4.27 show that as the number of DOF is increased the DG order of convergence is significantly better for purely smooth problems.

Comparison of the results of the Sod shock tube case for varying polynomial degrees using the DG method on a mesh with $n_e = 40$ to the FV method with refined meshes is plotted in Figure 4.28. As expected, when the mesh is refined, the resolution of the shock for the FV method is steeper and the solution resembles more closely the expected analytical solution for the Sod test case. If the gradient of the slope at the shock for $p_s = 8$ using the DG method is compared to the results for $n_e = 360$ of the FV method, it looks as if the FV method is steeper. This would suggest that the shock is better resolved in the FV method. Although this may be the case, the results from the FV method between the shock and the expansion fan, where the discontinuity is present, vary from the analytical solution. The two areas of constant density are in fact, lower than expected prior to the discontinuity and higher than expected after the discontinuity.

Figure 4.29 compares results obtained for the Shu-Osher case using the DG method on a mesh with $n_e = 96$ with

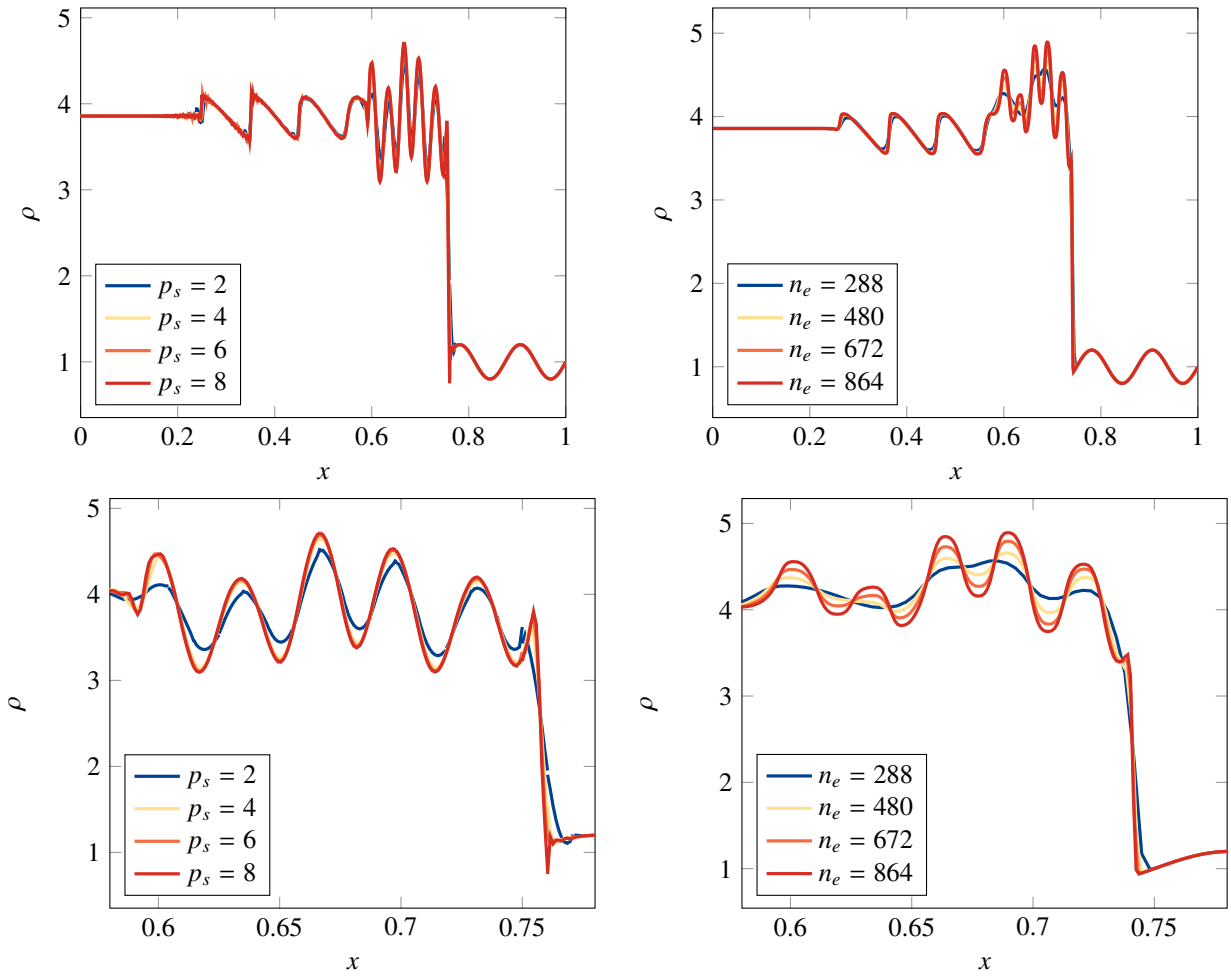


Figure 4.29. Density plots of the Shu-Osher test case for comparison of the DG method with $p_s = 2$, $p_s = 4$, $p_s = 6$ and $p_s = 8$ on a mesh with $n_e = 96$ (left) against the FV method on meshes with $n_e = 288$, $n_e = 480$, $n_e = 672$ and $n_e = 864$ (right).

varying polynomial degrees with the FV method for varying levels of mesh refinement. A much more significant change in the solution can be noted when comparing the mesh refinement of the FV method. For the coarsest mesh, the oscillatory region behind the normal sock is almost completely smeared and the oscillations are almost indistinguishable when using the FV method. There also appears to be a significant difference in the amplitude of the peaks in that region. Although the shock front, for the highest level of FV grid refinment, has a steeper gradient than the DG method with $p_s = 8$, comparing with results from [12, 17, 39, 40, 41] suggests the DG method appears to be in better agreement than the FV method. It is important to note that, contrary to the DG method, the FV method doesn't have regions with sporadic oscillations caused by the presence of higher order polynomials within elements.

The comparison of the density distribution for the interaction between two shock waves colliding together in the Woodward and Colella blast wave problem is presented in Figure 4.30. On the left are results using the DG method, and on the right, results using the FV solver. The center portion, where the interaction between the waves can be seen, is poorly resolved by the FV method, especially when compared to other published data in [12, 16, 17, 43]. Similarly to the other results containing normal shocks, as the grid is refined, the shocks are resolved more accurately and a steeper gradient can be seen between the pre and post shock regions.

For all one dimensional test cases, the FV method, at its highest level of grid refinement had a steeper resolution of the shock when comparing to the DG method with $p_s = 8$. On the other hand, for regions of the flow were the shock interacted in a complex manner, the FV solver did not give results which are comparable to previously published data. This indicates that the overall accuracy of the DG method, in smooth regions, is better than the FV method even if the shock front is not as steep.

The double Mach reflection is used as a two dimensional comparison between the DG and FV solvers because the FV solver is incapable of resolving unstructured grids. The mesh used for the DG simulations is a partially unstructured mesh, shown in figure 4.21. This corresponds to a total number of DOF of 135567 for $p_s = 2$ and 1220103 for $p_s = 8$. On the other hand, the mesh used for the FV solver is a fully structured mesh with a rectangular domain containing 200000 elements for the coarse mesh and 50 million elements for the refined mesh. Figures 4.31 and 4.32 show the comparison for these two meshes with the results from the DG method using $p_s = 2$ and $p_s = 8$. Figure 4.31 (a) is for $p_s = 2$ which appears to have similar results to Figure 4.31 (c), the coarse mesh of the FV solver. As the polynomial degree is increased in the DG method, a tendency similar to the one observed for mesh refinement in the FV method can be seen in the re-circulation area. Vortex like structures start to form along the primary slip line.

Based on the results above, it is apparent that the FV solver is better at handling discontinuities, while on the other hand the DG solver is significantly better at resolving the smooth areas of the flow. It is also interesting to observe that increasing the polynomial degree using DG acts the same as refining the mesh for FV. Depending on the type of problem being solved, DG or FV could be better suited for yielding a more accurate solution. In order to draw any definitive conclusions, additional testing is required. It is important to note that the FV solver used was designed and tailored

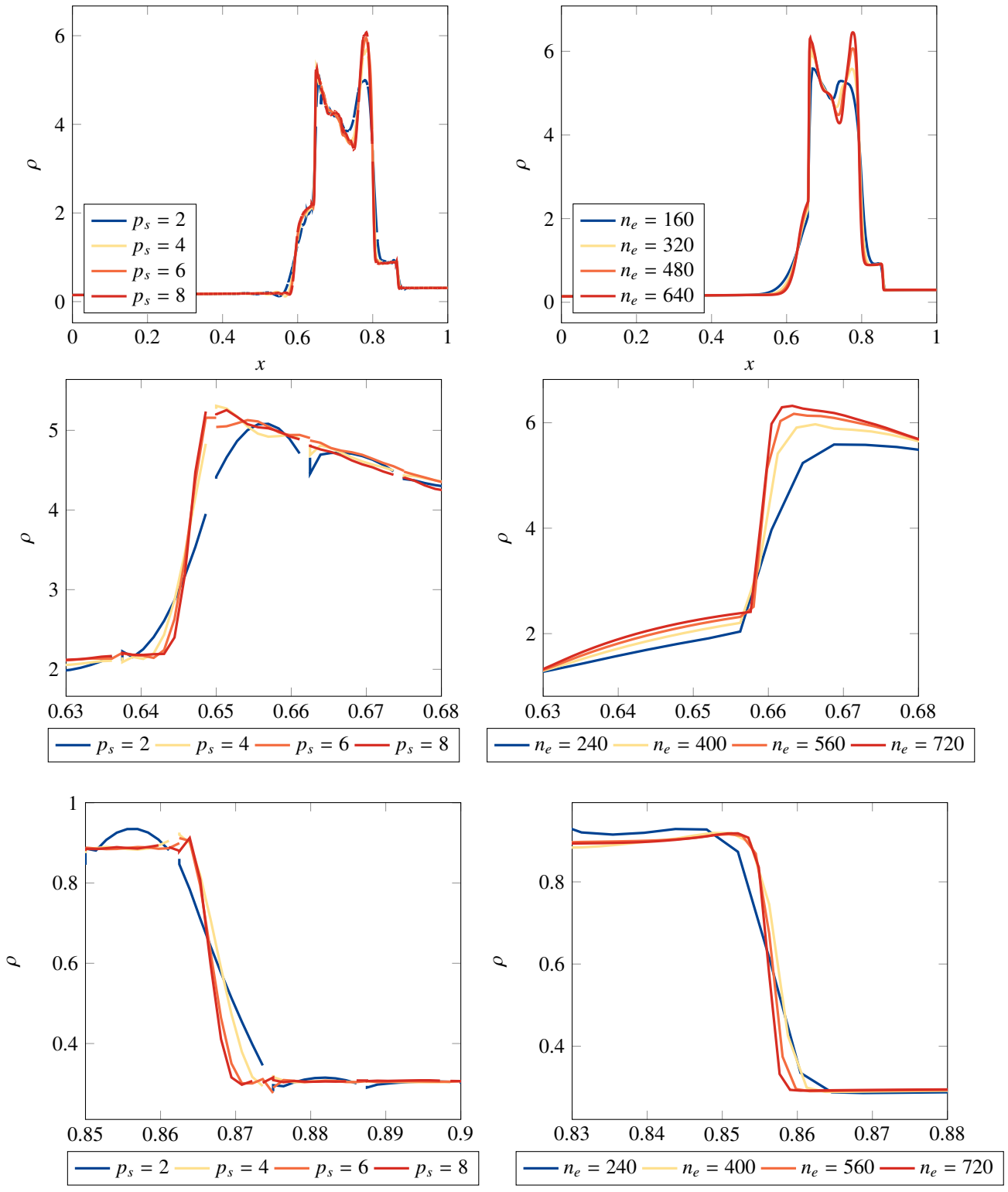
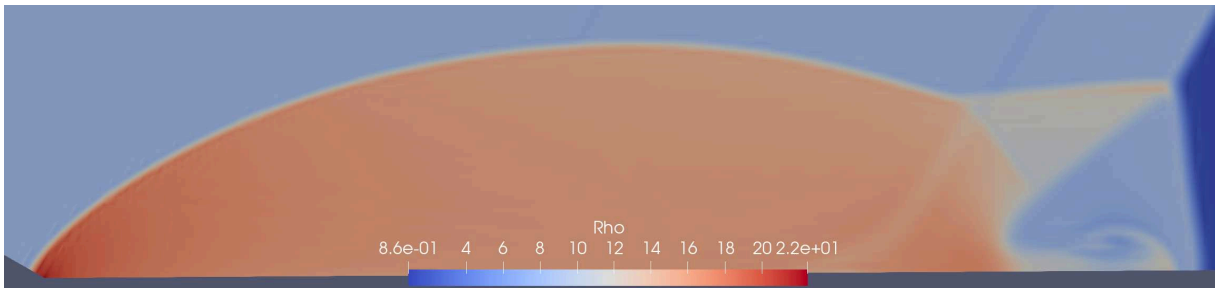
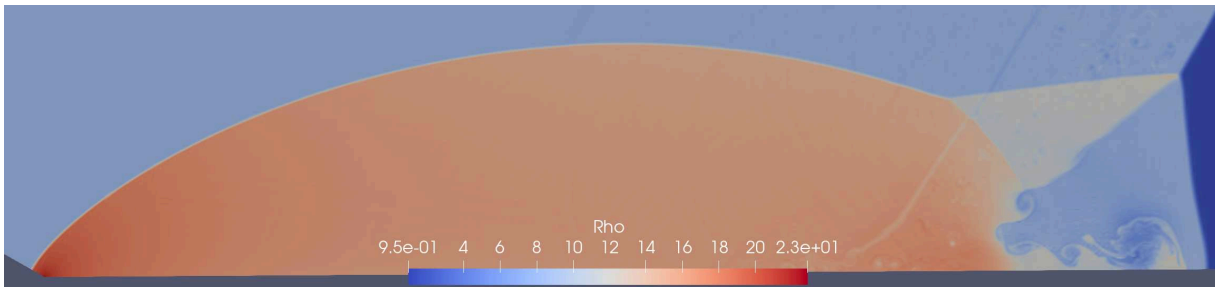


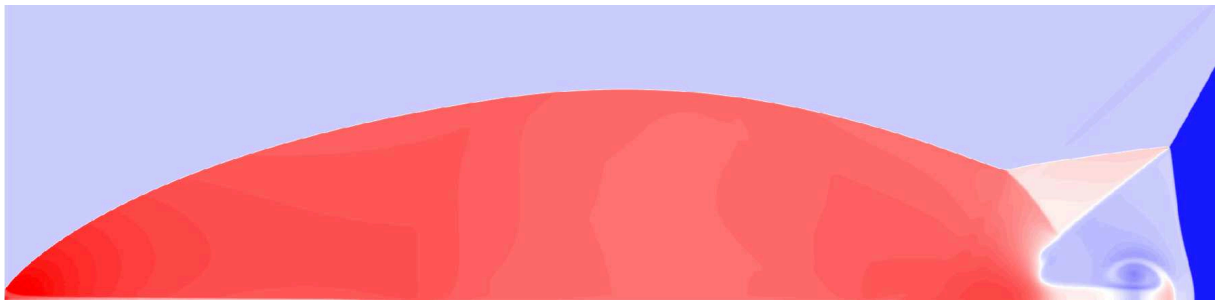
Figure 4.30. Density plots of the Woodward and Colella blast wave problem for comparison of the DG method with $p_s = 2$, $p_s = 4$, $p_s = 6$ and $p_s = 8$ on a mesh with $n_e = 80$ against the FV method on mesh with $n_e = 240$, $n_e = 400$, $n_e = 560$ and $n_e = 720$ (right).



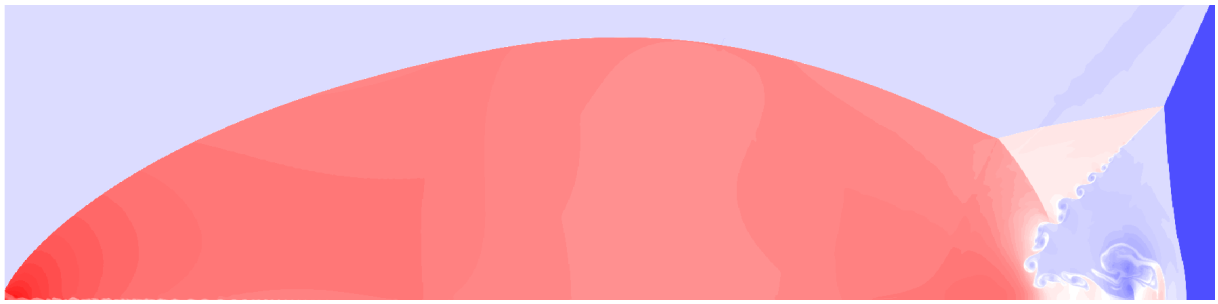
(a) DG with $p_s = 2$



(b) DG with $p_s = 8$

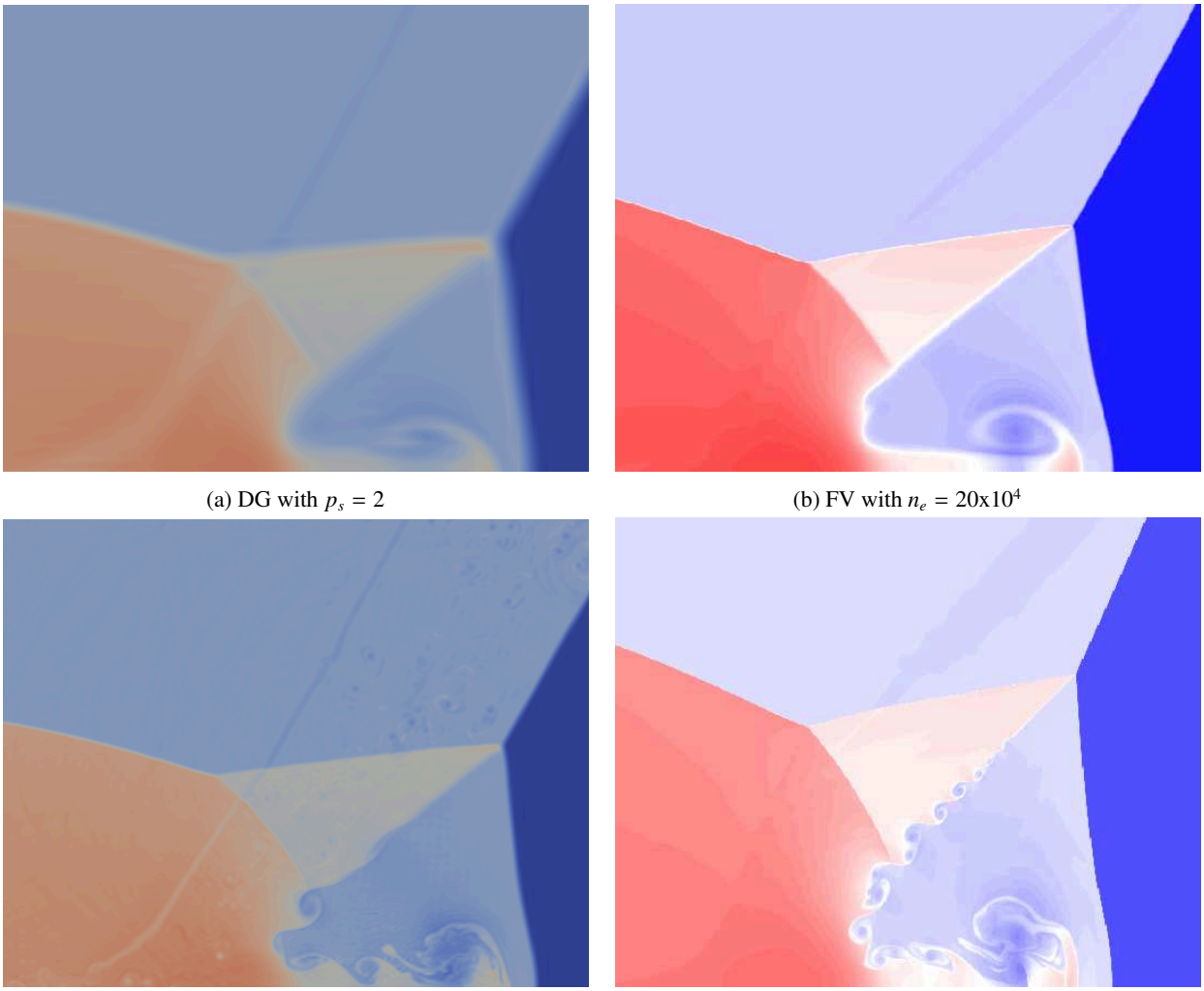


(c) FV with $n_e = 20\text{x}10^4$



(d) FV with $n_e = 50\text{x}10^6$

Figure 4.31. Density field for the double Mach reflection with varying polynomial degrees comparing the results for $p_s = 2$ and $p_s = 8$ using the DG method with the results from a coarse mesh and a mesh with 250 times more elements.



(a) DG with $p_s = 2$

(b) FV with $n_e = 20\text{x}10^4$

(c) DG with $p_s = 8$

(d) FV with $n_e = 50\text{x}10^6$

Figure 4.32. Close up of density in the vicinity of the primary slip line and Mach stem for the double Mach reflection comparing the results for $p_s = 2$ and $p_s = 8$ using the DG method with the FV results using a coarse mesh and a mesh with 250 times more elements.

mainly for problems with detonations and chemical reactions occurring in the domain.

Chapter 5

Adaptive Polynomial

Based on the strong correlation between increase in polynomial degree and better resolution of shocks for a mesh with a fixed number of elements, a polynomial adaptation routine is explored. Results in figures 4.6, 4.9 and 4.11 show three 1D test cases for which shocks are present in the domain: the Sod shock tube, the Shu Osher problem, and Woodward and Colella blast. In all three cases, as the polynomial degree is increased, the solution resolution around the shock is significantly better. A sharper jump in the given conserved variable, covering fewer elements, is obtained. Additionally, it should be noted that the proposed dilation based shock capturing technique, with the artificial viscosity in equation 3.20 using function parameter values $\alpha = 1, \beta = 0.1$ and $k_h = 1$, have proven to be effective at detecting regions with shocks and mitigating the presence of large oscillations associated with the polynomial constructed to resolve them. Interestingly, in the case of the Sod shock tube in figure 4.6, and the Woodward and Colella blast wave in figure 4.11, the oscillations for higher polynomial degrees appear to be smaller than for lower order polynomials. Visually, it appears that the approach combining the dilation shock sensor with an artificial viscosity function similar to that proposed by Persson and Peraire provides good shock resolution which improves with increased polynomial degree.

By comparing the results on a uniform mesh for varying polynomial degrees, a clear advantage can be observed and even more so around shocks. Based on this observed advantage, additional simulations were conducted to determine if varying the polynomial degree in the vicinity of a shock could have a similar effect to refining the mesh in order to get a sharper solution. In order to determine the validity of this approach, the three one-dimensional test cases for which polynomial refinement and mesh refinement were performed are once again tested for polynomial adaptation on a fixed grid. The polynomial adaptation approach consists of detecting regions in which the shock is present using the dilation shock sensor, increasing the polynomial degree for that cell as well as adjacent ones in order to help increase shock resolution. Computationally, this is significantly less expensive since higher order polynomials are only being used in regions where the shock is present. Typically, only a small portion of the domain is covered. On the majority of the domain, where the solution is smooth, a lower order polynomial is used.

5.1 Sod Shock Tube

The Sod Shock tube test case has two expansion fans, a contact discontinuity and a normal shock which makes it an ideal case to see if the p_s adaptation routine is capable of resolving regions in a similar manner to a routine which would typically utilize mesh refinement instead. Figure 5.1 shows the results for the simulations using different levels of p_s refinement where artificial viscosity is added. As was the case for the uniform polynomial degree results shown in figure 4.7, as the adaptive polynomial degree is increased so the resolution of the shock gets greater and the solution becomes more accurate. It is also important to note that when comparing the results from simulations with uniform polynomial degree and adaptive polynomials, the shock resolution quality is nearly indiscernible between the two. This indicates that the adaptive routine is capable of accurately resolving the shock while maintaining fewer average degrees of freedom when compared to the uniform polynomial routine. Figure 5.1 (e) shows the adaptive routine with $p_s = 2$ and $p_s = 8$ as the adaptive polynomial. It can be seen that the higher order polynomial is present in the vicinity of the shock and the corners of the expansion fan only. Table 5.1 shows the ratio of the number of degrees of freedom when compared to the uniform $p_s = 2$ case for $n_e = 40$ 80 and 160 elements. Since the shock and corners of the expansion fan are limited to a small number of elements, the number of cells in which a higher polynomial degree is required is very low and this is independent of the total number of elements. As the mesh is refined the number of elements requiring higher order polynomials doesn't increase proportionally. This means that when using finer meshes the computational cost of using the adaptive polynomial routine is minimal, it is estimated to increase the total degrees of freedom by $\sim 10\%$, while giving a much higher shock resolution.

Table 5.1. Ratio of number of degrees of freedom for the adaptive Sod shock tube cases relative to uniform $p_s = 2$.

p_s	$n_e = 40$	$n_e = 80$	$n_e = 160$
$2, 2$	1.00	1.00	1.00
$2, 4$	1.15	1.07	1.04
$2, 6$	1.33	1.12	1.08
$2, 8$	1.45	1.15	1.13

5.2 Shu-Osher

The Shu Osher case represents a more complex interaction between a normal shock and a sinusoidal density distribution. This idealized one-dimensional shock-turbulence interaction is a more difficult problem to resolve which requires a more robust solver capable of handling these interactions. The adaptive polynomial method is tested on a mesh with $n_e = 96$ figure 5.2 compares the uniform $p_s = 2$ solution with the adaptive $p_s = 2, 4$, $p_s = 2, 6$ and $p_s = 2, 8$ solution. In all cases, as the adaptive polynomial degree is increased the shock resolution also increases. In figure 5.2 (e) it can

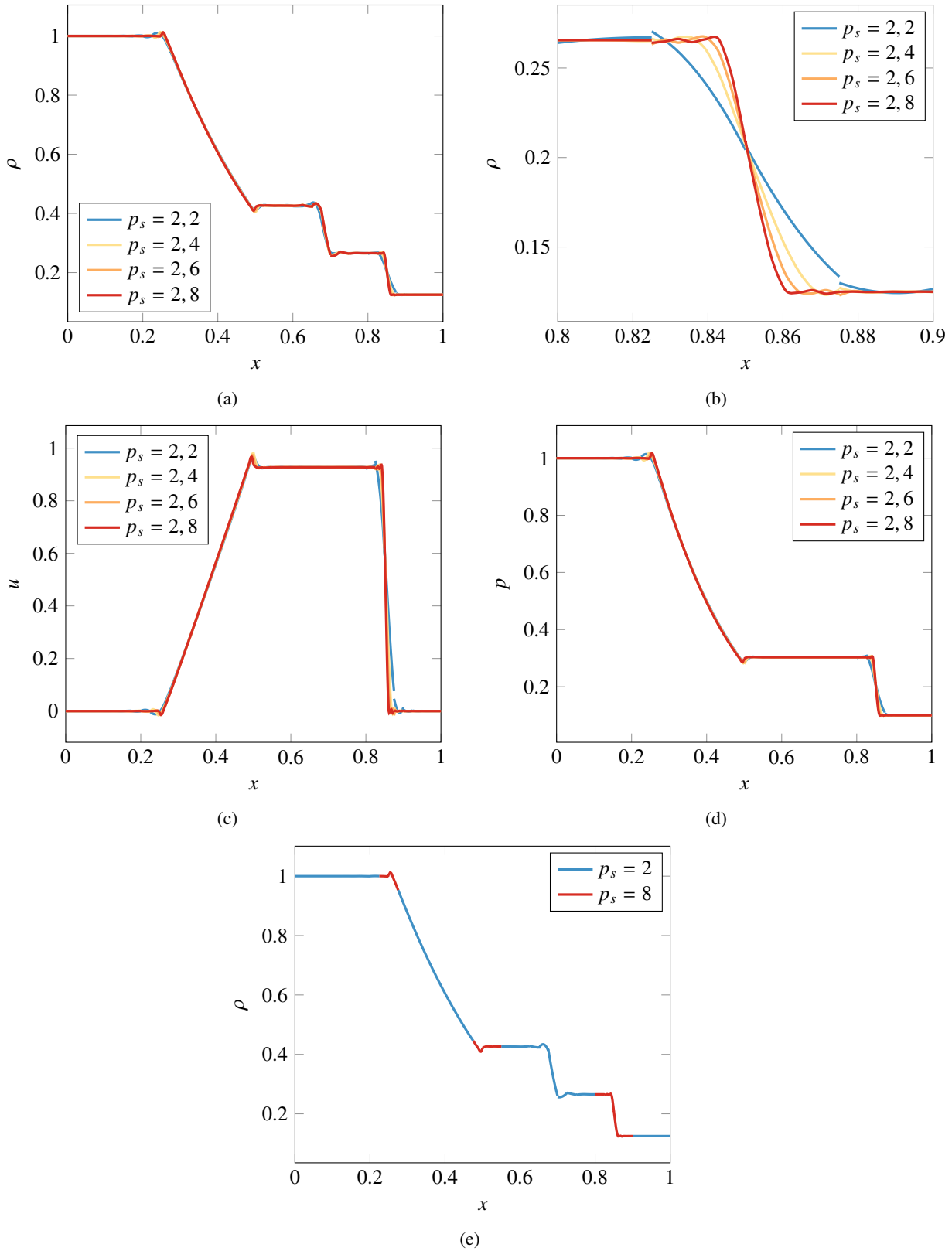


Figure 5.1. Plots of density (a, b), velocity (c), and pressure (d) for Sod's shock tube at $t = 0.2$ using adapted solution polynomials of degree $p_s = 2$ to 8. (e) shows the distribution of $p_s = 2$ (blue) and $p_s = 8$ (red) elements for the adaptive $p_s = 2, 8$ simulation.

be seen that the adaptation scheme is increasing the polynomial degree at the normal shock traveling to the right of the domain as well as the weaker shocks being formed traveling to the left of the domain behind the oscillatory region. This gain in accuracy, while keeping the overall degrees of freedom low, suggests that substantial gains in accuracy are possible while keeping computational costs low.

5.3 Woodward and Colella Blast

The results of uniform $p_s = 2$ and adaptive $p_s = 2,4$, $p_s = 2,6$ and $p_s = 2,8$ on a mesh with $n_e = 80$ and shown in figure 5.3. Once again, the increase in polynomial degree gives better shock resolution while only applying the increase in polynomial degree at the locations where shocks and expansion fans are present. By simply increasing the polynomial degree in zones where there are shocks, a significant gain in precision in comparison to a uniform $p_s = 2$ can be observed. On the other hand when comparing the different adaptive polynomial degrees, there is only a slight difference in the steepness of the shock being resolved.

For all three one dimensional adaptation routines there is a clear gain in accuracy as the polynomial is increased in regions where the dilation based shock detector adds artificial viscosity. Even adaptive routines with the degree changing from $p_s = 2$ to $p_s = 4$ greatly increases the steepness of the resolved shock. This allows for more accurate solutions while maintaining the average number of degrees of freedom low. Consequently, the computational cost of increasing the polynomial degree in regions where shocks are present remains low and overall accuracy of the solution is increased significantly. A drawback of this method is one that is also present for adaptive mesh refinement, specifically if a region where no shocks are present is indicated as containing shocks the polynomial degree will increase even if it is not required. Just as in adaptive mesh refinement, if this is the case the accuracy of the solution in this region is not diminished. Only additional computational costs ensue.

For all three one-dimensional test cases for which adaptive polynomial routines were used, it can be observed that adaptive routines have results similar to their uniform polynomial order counterparts. A uniform $p_s = 8$ yields solutions very similar to an adaptive $p_s = 2,8$ with a much larger average number of degrees of freedom per element. This suggests that the adaptive polynomial routine can yield results similar to mesh refinement. It would also be possible to combine the two adaptive methods to give even better results.

5.4 Double Mach Reflection

The true advantage of the adaptive polynomial comes when the number of elements is significantly increased and nothing does this like adding an additional spatial dimension. For this reason, the adaptive polynomial routine was tested on the 2D double Mach reflection to determine its ability to reduce computational costs while adding higher accuracy

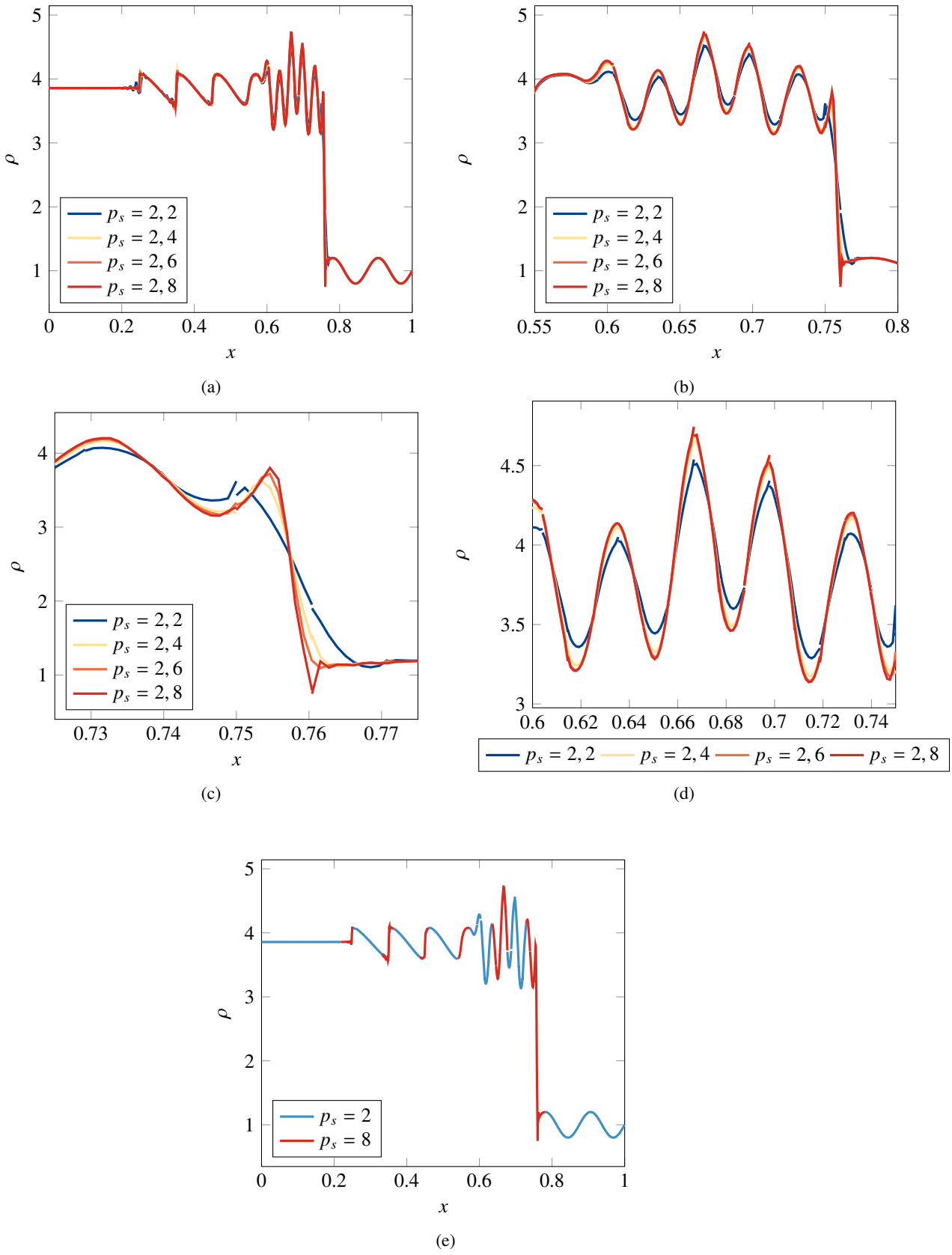


Figure 5.2. Shu Osher problem. Density (a, b, c, d) is shown for different sections of the flow field. The location of the polynomial refinement is shown in (e) with polynomial adaptation for 96 element mesh.

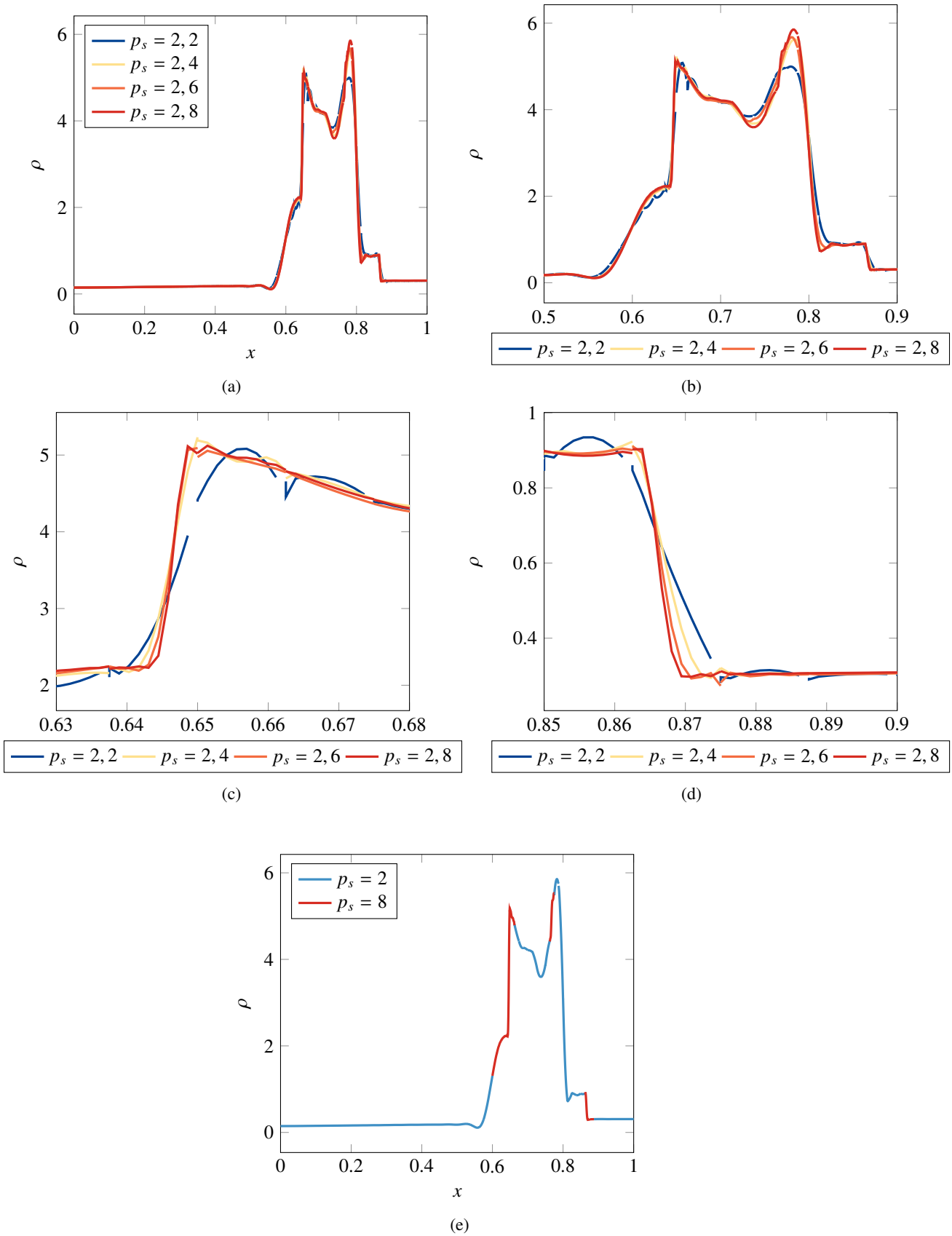


Figure 5.3. Woodward and Colella blast problem. Density (a, b, c, d) is shown for different sections of the flow field. The location of the polynomial refinement is shown in (e) with polynomial adaptation for 80 element mesh.

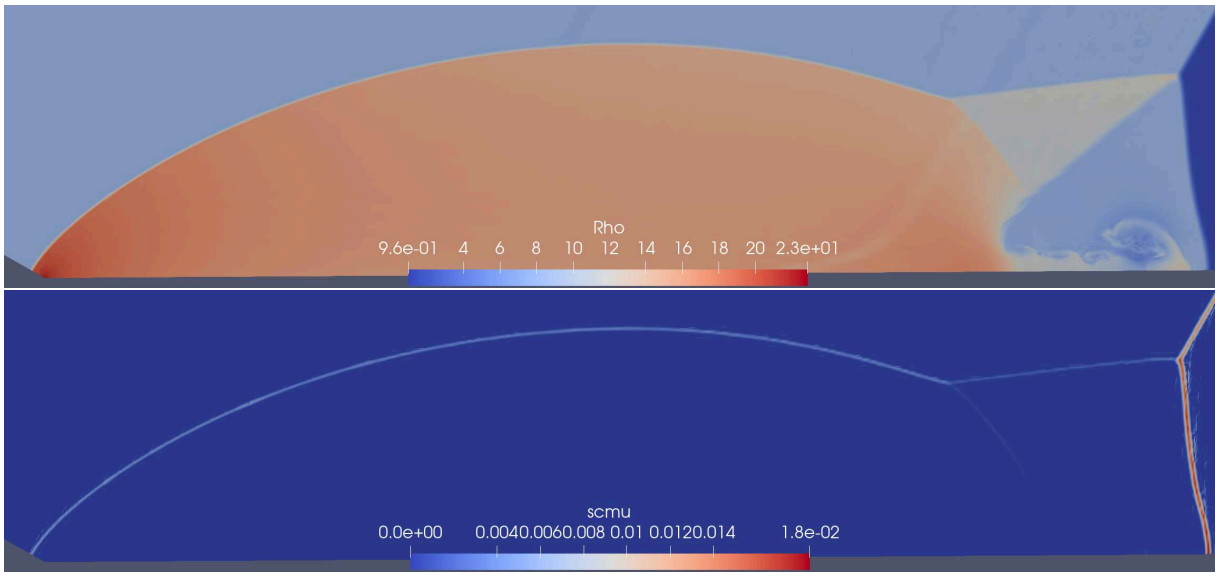


Figure 5.4. Density and artificial viscosity for the double Mach reflection with adaptive polynomial degrees.

than the lower polynomial simulations. Figures 5.4 and 5.5 show the density and artificial viscosity results for adaptive $p_s = 2$ to $p_s = 7$ where shocks are present. Attempts to run different adaptive polynomial schemes were conducted, but stability issues with different values of the adapted polynomial were encountered.

Figure 5.4 and 5.5 show the results for the $p_s = 2$, $p_s = 7$ adaptive simulation at $t = 0.2$. When comparing with the results for $p_s = 2$ in figures 4.22, 4.23, 4.24 and 4.25, it is clear that the adaptive scheme resolves the shocks without smearing them like the uniform $p_s = 2$ scheme does. The presence of the vortex like structures at the jet are also distinguishable as in the results for the uniform $p_s = 8$ simulation. The presence of these structures in the adaptive results for $p_s = 2$ to $p_s = 7$ shows that this adaptive routine could give better results than running a uniform $p_s = 6$ simulation while maintaining a significantly lower average number of degrees of freedom. The thickness of the regions where artificial viscosity is added at the incident shock wave, primary Mach stem, primary reflected shock wave secondary reflected shock wave and secondary Mach stem are also thinner than for the uniform $p_s = 2$ simulations.

In Figure 5.4 regions in blue indicate where the polynomial element is $p_s = 2$ and regions in red indicate $p_s = 7$ at $t = 0.2$. The equation 3.16 which is used for detecting the presence of shocks and subsequently adding artificial viscosity is also used to increase the polynomial degree. It can be seen that behind the triple point, where the incident shock wave, primary Mach stem and primary reflected shock wave meet, there appears to be ripples traveling outward were shocks are detected and the polynomial degree is increased. Additionally, trailing behind and parallel to the incident shock wave is a secondary numerical wave that is detected as a shock and the polynomial degree is also increased in this region. These regions where shocks are detected but not physically present could be the reason for instabilities in the other adaptive combinations.

All test cases where the adaptive routine was used gave a more accurate solution than their uniform counterparts

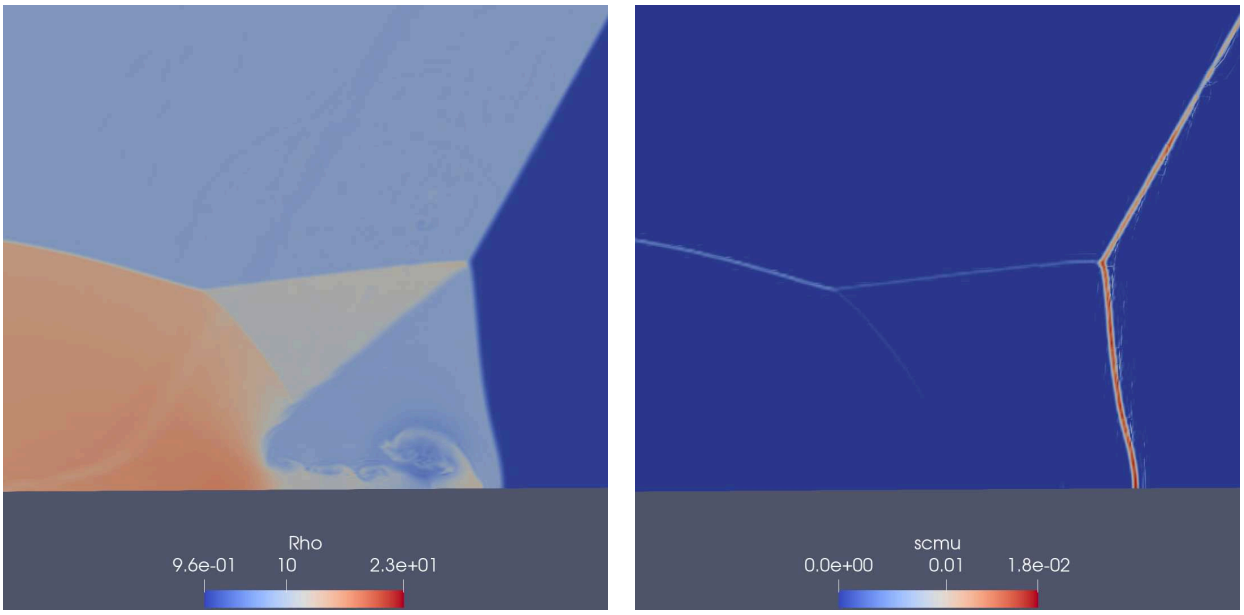


Figure 5.5. Close up of density and artificial viscosity in the vicinity of the primary slip line and primary mach stem for the double Mach reflection with adaptive polynomial degrees.

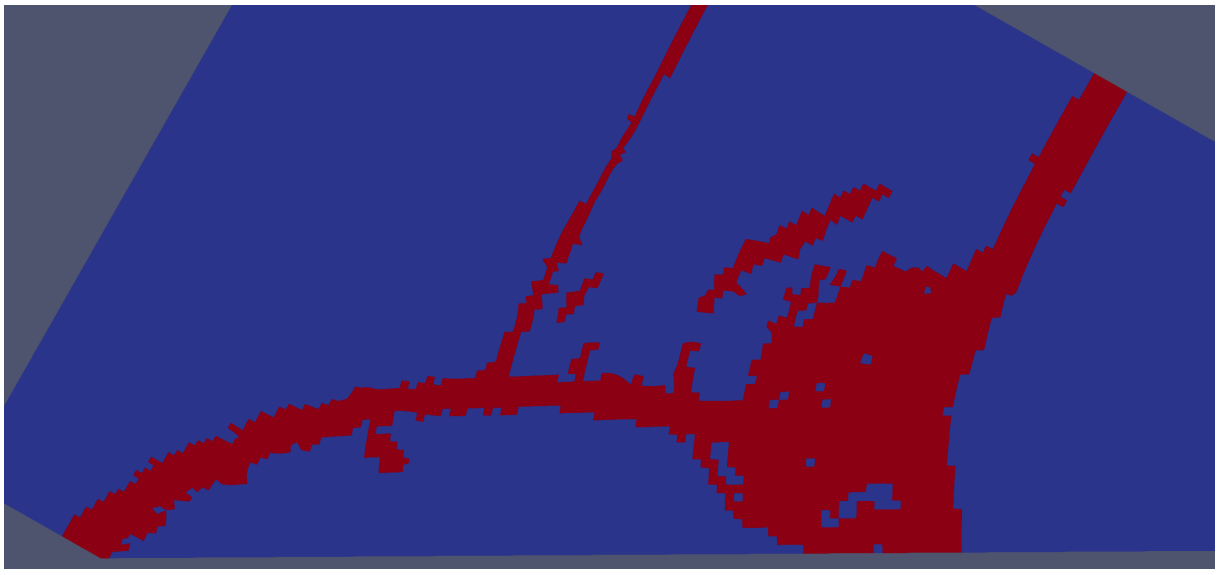


Figure 5.6. Overall view of cells with $p_s = 2$ in blue and $p_s = 7$ in red for adaptive polynomial degrees.

while maintaining a lower average number of degrees of freedom. This suggests that a more accurate solution can be obtained by utilizing a low value of p_s in smooth regions and increasing p_s in regions where shocks are detected.

Chapter 6

Real World Case

In addition to comparing numerical simulations to problems tested by other authors the ability, for the HORUS solver to capture real world phenomena numerically was put to the test by comparing it to experimental data acquired in the lab. The experiment was conducted in the context of another student's master's research into the interaction between shock waves and turbulence. In order to create a scenario where a shock wave would interact with a turbulent region, an apparatus was designed in which a high pressure jet of air is injected for a fixed amount of time. This injection is at the middle of the tube where a turbulent region forms and the overall pressure inside the tube increases. At the end of injection, at one end of the tube, an electric igniter is activated which increases the energy in that small section of the testing apparatus, in turn creating a shock wave. As the shock wave travels down the tube, it eventually crosses the viewing section where the jet of air was injected and the turbulent region. In order to visualize the interaction between the two a Z-type schlieren visualization system was set up to acquire pictures of the density gradient in the viewing area.

Using the schlieren pictures taken during the course of the experiment, it is possible to build a two dimensional simulation of the experimental set-up and then determine if similar interactions are occurring in the numerical solution. Without any exact analytical solution to the interaction between turbulence and shocks, the only way to validate the ability for the solver to capture real world phenomena is to compare its numerical results to experiments such as these. Although only two dimensional simulations for the tube were conducted, comparison with the images acquired still offers valuable insight on how HORUS can handle complex two dimensional interactions. In order to get a better grasp of what is occurring, the full Navier-Stokes equations are used to simulate more accurately the turbulence formed as fluid is injected into the pipe and subsequently interacts with a moving shock.

The simulation parameters used to compare the numerical results with experimental data were a Reynolds number of 6000 and 150000, Prantl number of 2.71, a polynomial $p_s = 3$, injection times of $\Delta t = 50$ and $\Delta t = 100$ at a ratio of injected to ambient fluid pressure of 2:1. In all cases the increase in internal energy at the far right side of the domain was a 3:1 increase which occurred $0.1\Delta t$ after injection was completed. Additionally, after injection, the inlet boundary

(a) Poiseuille (b) Turbulent (c) Flat

Figure 6.1. Density comparison of Poiseuille, turbulent and flat injection profiles for mesh with $n_e = 224756$ at $Re = 2000$.

condition was converted to a wall boundary condition. The mesh was fully structured, consisting of 224756 square elements in 2D. All simulations were conducted with the positive limiter turned on and dilation based shock capturing enabled.

6.1 Injection Phase

Prior to comparing with the experimental data, the need to determine the flow conditions at the inlet of the plenum was required. Three possible inlet conditions where explored. A Poiseuille flow distribution, a turbulent flow distribution and a flat distribution. It was determined by qualitatively comparing the numerical results that very little difference can be seen between the Poiseuille and turbulent types of injection however the flat injection profile differs significantly from the other two. The main difference lies in the shape of the fluid as it initially traveling through the injection tube. Comparison of all three injection profiles are shown in figure 6.1 for the transition time from when the inlet is opened and moments before the bulk of the fluid exits the tube. The Poiseuille injection profile was used for all test case scenarios in order to facilitate calculations. This in turn allows for better control of the average injection velocity,pressure and mass flow rate.

In addition to determining the required injection profile the consistency of the solution obtained, prior to the shock being generate, was compared by running preliminary simulations on varying levels of mesh refinement. As the mesh was refined the solutions density distribution shown in figure 6.2 appears to visually follow a trend towards the densest mesh. In addition, the largest variation appears to be in the plum of fluid exiting the injection tube. The general structure and scale of the turbulent area to the left and right of the tube are similar when compare in figure 6.2.

After determining the injection profile and the mesh to be used for the simulation, varying injection times were tested to determine the amount and scale of the turbulence being created in the viewing section. Although the tests run in the lab do have varying injection times it was impossible to determine what the turbulence looked like in the

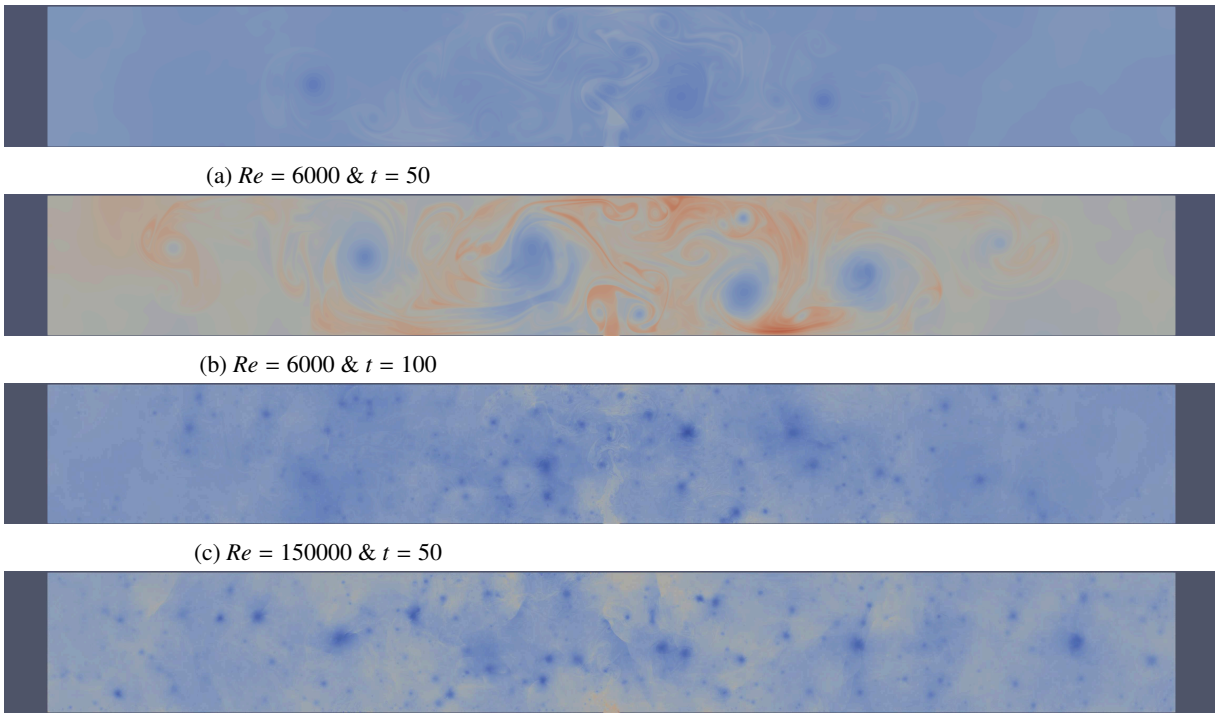
(a) $n_e = 13842$

(b) $n_e = 55688$

(c) $n_e = 127055$

(d) $n_e = 224756$

Figure 6.2. Density comparison of the solution for injection time of $t = 50$ with Poisseuille injection profile for mesh with $n_e = 13842$, $n_e = 55688$, $n_e = 127055$ and $n_e = 224756$ at $Re = 2000$. Used to determine if different levels of mesh refinement would yield similar results prior to turning off the injection inlet condition.



(a) $Re = 6000$ & $t = 50$

(b) $Re = 6000$ & $t = 100$

(c) $Re = 150000$ & $t = 50$

(d) $Re = 150000$ & $t = 100$

Figure 6.3. Density comparison of the solution after continuous injection for $t = 50$ and $t = 100$ with Poisseuille injection profile for meshes with $n_e = 224756$ at $Re = 6000$ and $Re = 150000$.

tube before, during and after the shock was created by the spark plug. For this reason determining what kind of effect injection would have on the turbulence within the tube was investigated and comparison of two injection times can be seen in figure 6.3 for $Re = 6000$ and $Re = 150000$. The increase in injection time appears to increase the overall pressure in the tube, as is expected, when looking at the results from the experimental data. It also appears that the turbulent area around the injection site spreads further down the tube as injection time is increased. While the overall turbulence scales appears to remain unaffected by the increase in injection time. The greatest effects on the turbulent structure comes from the change in Reynolds number, at Reynolds number over 2000 according to [50] turbulence is present. Based on the equation 6.1 for ρ_j, U_j and D being the jet inlet density, velocity and diameter respectively.

$$Re = \frac{\rho_j U_j D}{\mu} > 2000 \tag{6.1}$$

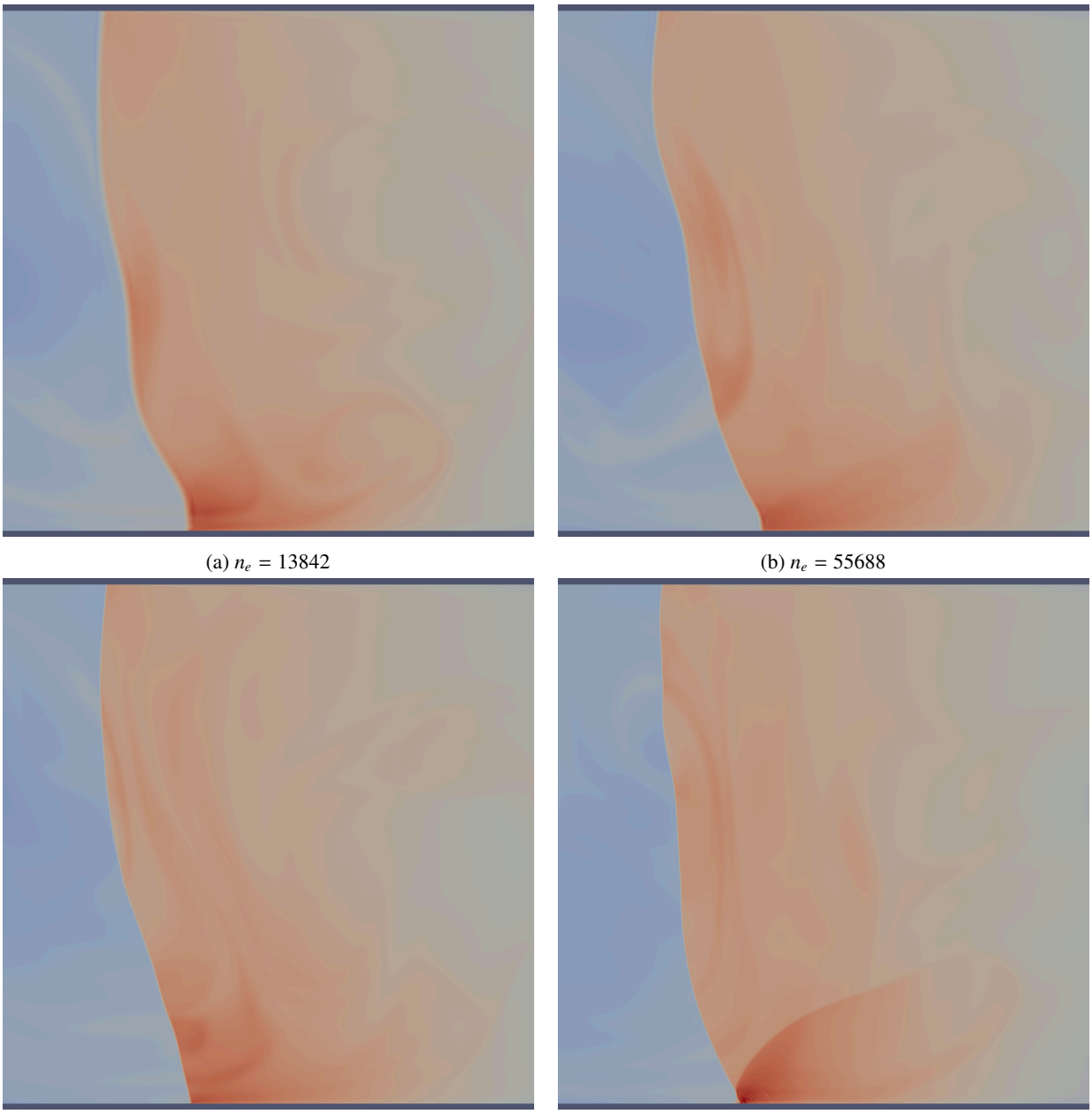
6.2 Shock Phase

Without any analytical or previous numerical results to base the simulations on conclusions about the exactitude of the results obtained were only speculative in nature and qualitative comparison between various simulations were conducted. In the numerical simulations, after a high pressure fluid was injected for a set period of time the source was cut off and the boundary condition at the inlet was changed to a wall. A pre-determined period of time was set before a shock was generated to the far right of the domain. To generate this shock the specific internal energy of a small portion of the domain was tripled which generated the desired shock. Initial investigation was conducted on ensuring that the mesh used was capable of capturing the fluid interaction between the shock and the turbulent region without causing the solver to crash or give unrealistic results. Mesh refinement comparison for the shock traveling through the domain can be seen in figure 6.4 and 6.5.

Comparison of the shock as it travels through the domain for varying levels of mesh refinement in figures 6.4 and 6.5 indicate that the general shape the shock front as it travels through the domain converges towards a general solution. Varying the level of grid refinement was observed, in figure 6.2, to change the exact turbulent structure, in turn this will also affect the way the shock from behaves. None the less, the general behavior of the shock, as it travels through the domain, appears to be consistent for varying levels of grid refinement. This indicates that although no exact conclusion can be drawn a general idea of how the shock interacts with the turbulent structure can be determined if the most refined mesh is used for further simulations.

Results in figures 6.6 and 6.7 show the shock front before and after the injection tube for $Re = 6000$ and $Re = 150000$ on a mesh with $n_e = 224756$ for comparison with experimental results obtained. The results from the numerical simulation clearly indicated, as would be expected, that as the shock moves through the domain it is being weakened likely because of the friction on the walls and some energy dissipation caused by the vortices. The structure of the shock front warps as it travels through the turbulent areas in the direction of rotation of the vortex it interacts with. This explains why the simulations with the highest Reynolds number result in a shock front that is more distorted than the lower Reynolds number simulation. The numerical results also indicated that as the shock travels through the domain small turbulent structures start to form at the wall in the wake of the shock, for the higher Reynolds number simulation the turbulence at the wall start earlier in the domain.

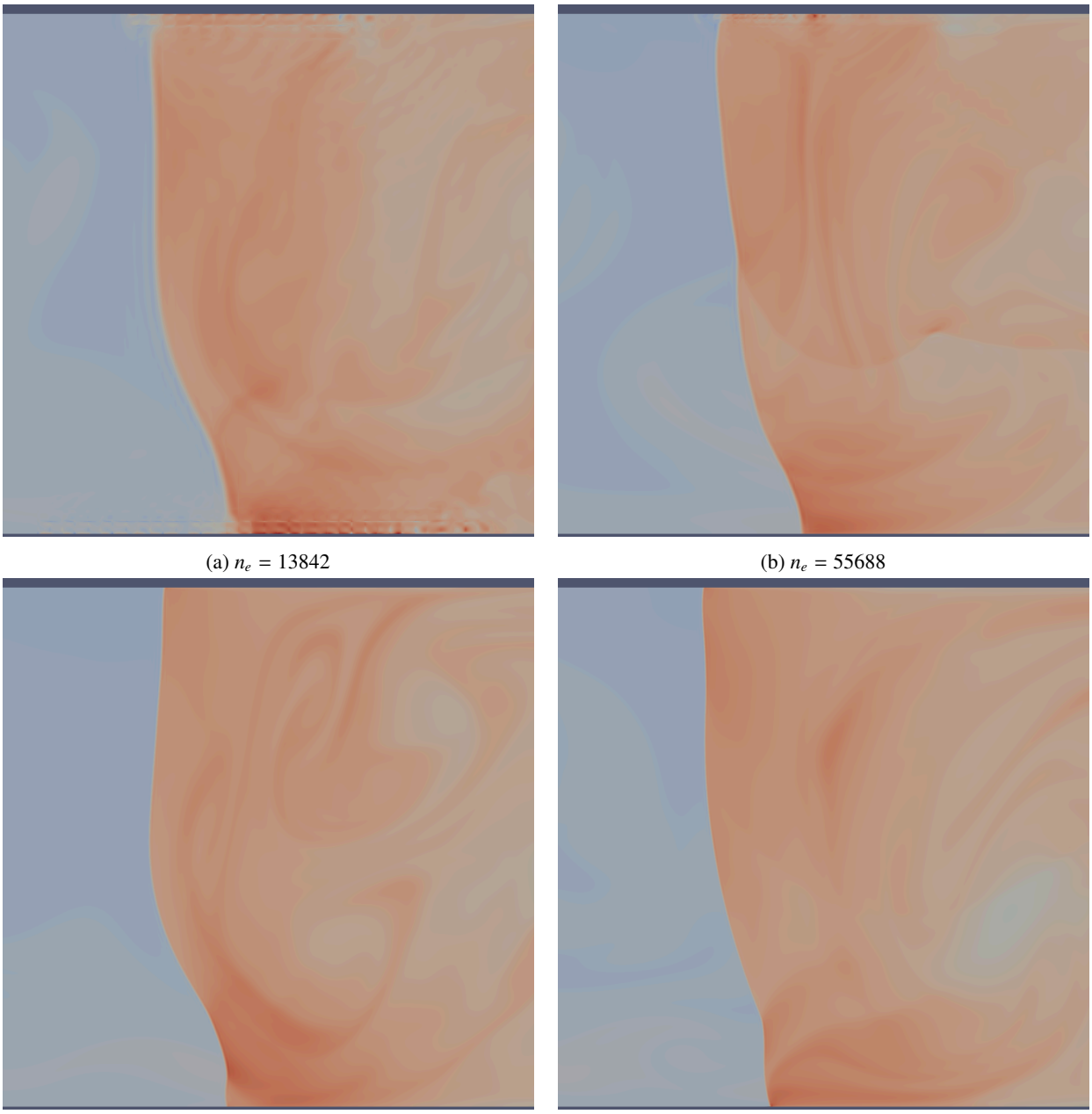
Based on the above comparison the average results from the interaction between the shock and the turbulent areas follow a similar trend. Although no definitive conclusions can be drawn from these simulations it appears that qualitative observations can be made. By comparing results between grid refinement and changes in Reynolds numbers that as the shock passes over turbulent area the front is deformed based on the direction and scale of the vortex structure. It can also be observed that the strength of the shock diminishes as it travels through the domain which is likely caused by friction with the walls and interactions with the turbulence.



(a) $n_e = 13842$ (b) $n_e = 55688$

(c) $n_e = 127055$ (d) $n_e = 224756$

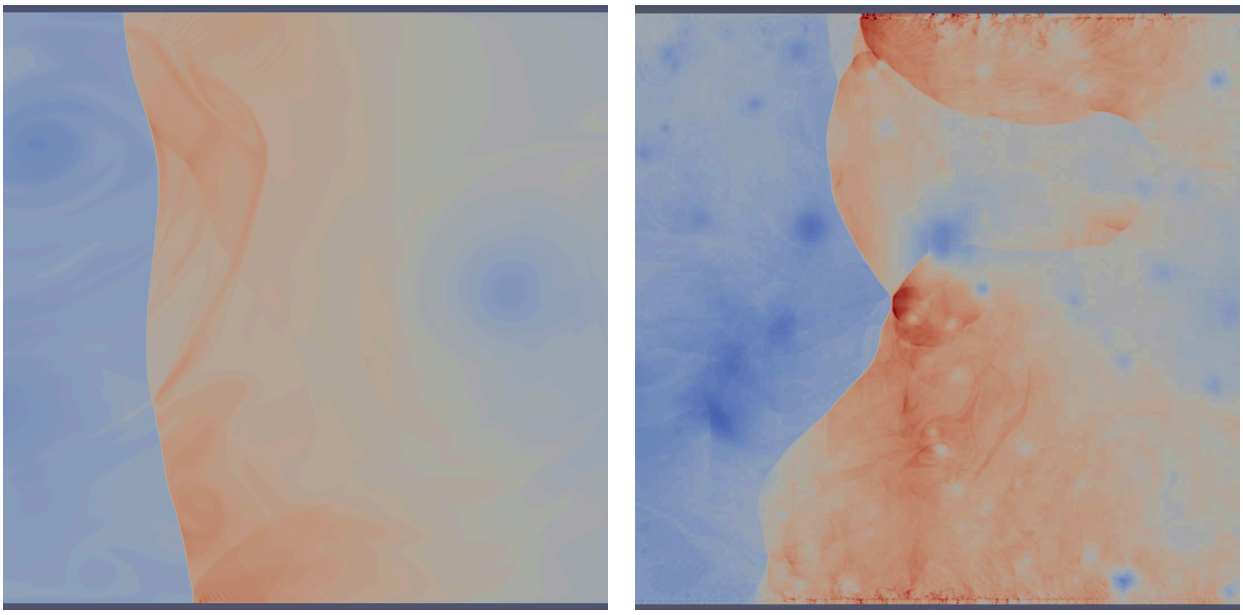
Figure 6.4. Density comparison of the shock front before it reaches the injection tube at the middle of the domain for meshes with $n_e = 13842$, $n_e = 55688$, $n_e = 127055$ and $n_e = 224756$ at $Re = 2000$.



(a) $n_e = 13842$ (b) $n_e = 55688$

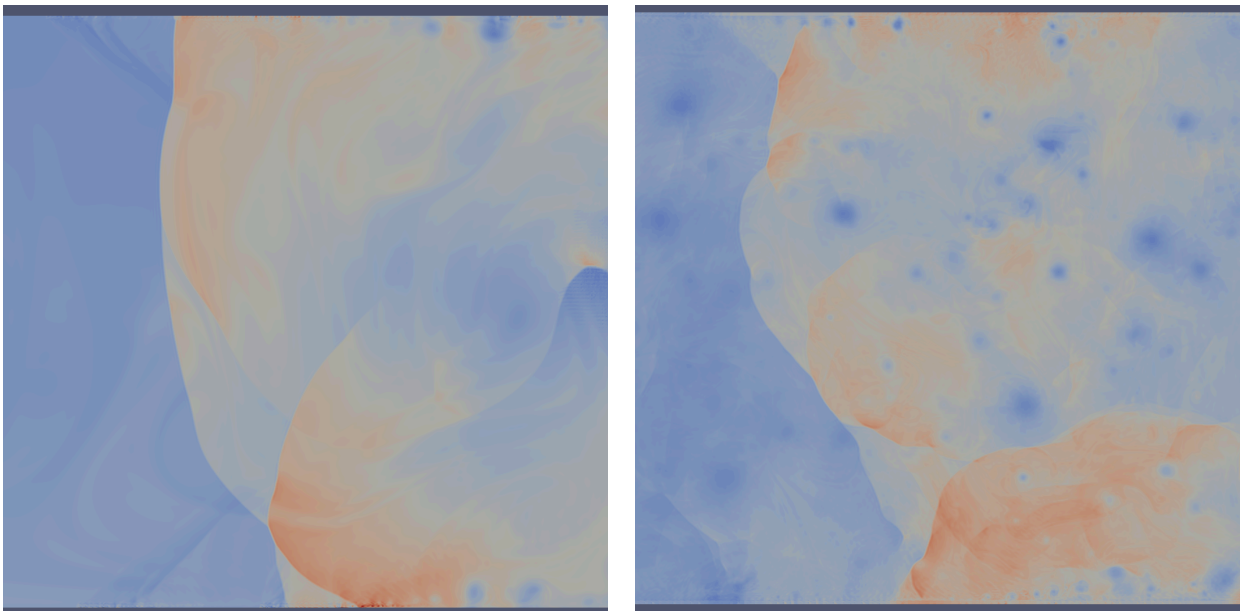
(c) $n_e = 127055$ (d) $n_e = 224756$

Figure 6.5. Density comparison of the shock front traveling through the domain after it has passed over the injection tube at the middle of the domain for meshes with $n_e = 13842$, $n_e = 55688$, $n_e = 127055$ and $n_e = 224756$ at $Re = 2000$.



(a) $Re = 6000$ (b) $Re = 150000$

Figure 6.6. Density comparison of the shock front before it reaches the injection tube at the middle of the domain for $Re = 6000$ and $Re = 150000$ and $n_e = 224756$.



(a) $Re = 6000$ (b) $Re = 150000$

Figure 6.7. Density comparison of the shock front traveling through the domain after it has passed over the injection tube at the middle of the domain for $Re = 6000$ and $Re = 150000$ and $n_e = 224756$.

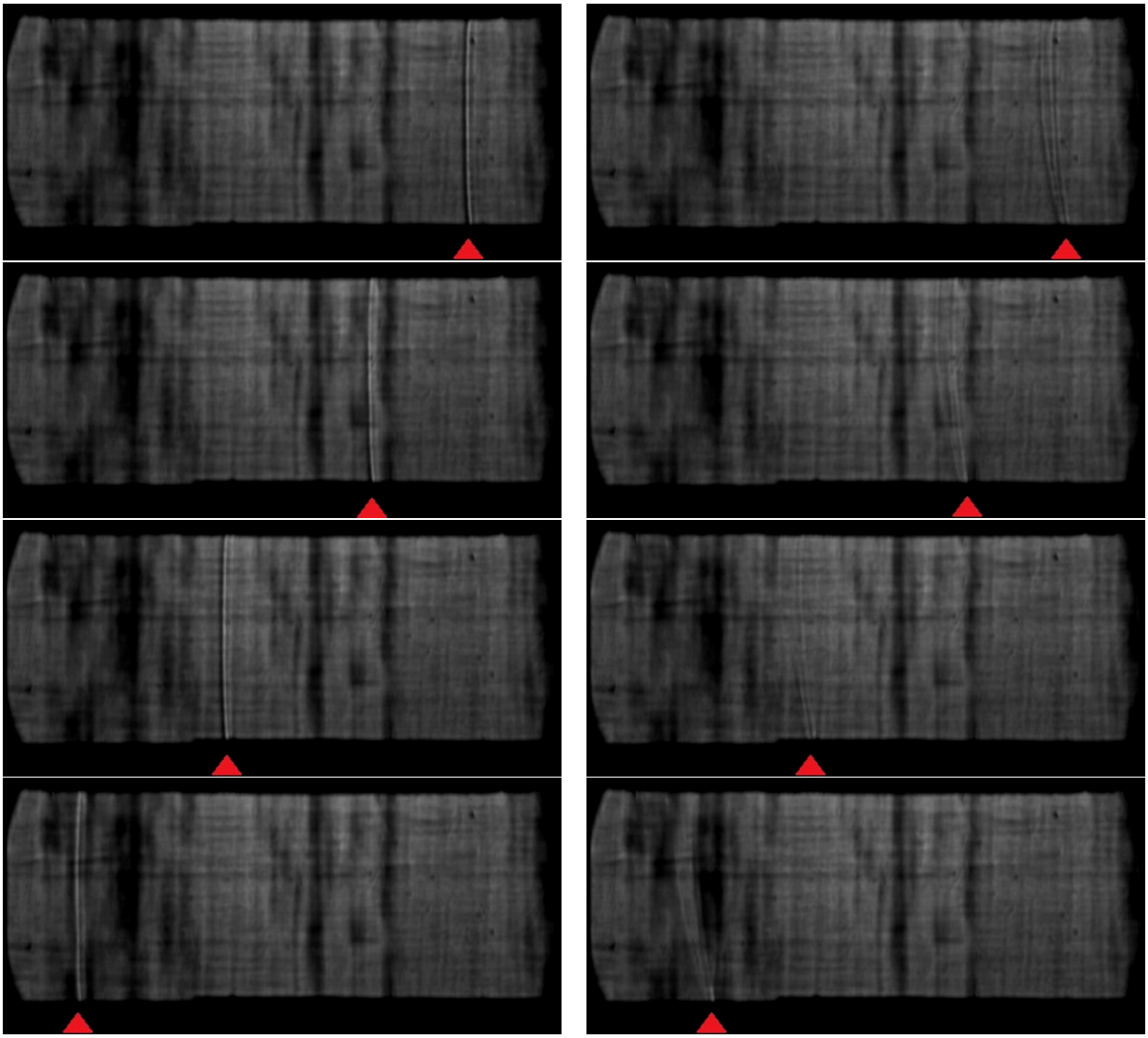
6.3 Experimental and Numerical Comparison

In order to further determine the effectiveness of HORUS, comparison of the numerical results obtained using a mesh with $n_e = 224756$ with the experimental results obtain in the context of [51] was conducted. Experimental results into the investigation of the interaction between shock waves and turbulence were determined using a Z-type schlieren visualization system which allows to capture images of the density distribution within the test tube while the shock is traveling through it. Based on the dimensions and operating conditions of the set up an approximate Reynolds number of = 1500000 and Prantl number of 2.47 was found and used for the numerical simulations. During the physical experiment various pressure ratios where tested for the initial pressure and injection pressure with a fixed final pressure within the tube. In the case of the numerical experiment the pressure ratio between the tube initial pressure and inlet pressure were fixed at 1:2 and the injection time was varied between $\Delta t = 50$ and $\Delta t = 100$ with a shock being generated by tripling the specific internal energy of the right most area of the tube after injection was completed. Since it was experimentally impossible to quantify turbulence prior to or after the shock had passed through the tube and only 11 frames are obtained per run only qualitative comparison of the results are investigated.

The images of the density distribution initial tube pressure of $100kPa$ and an injection pressure of $200kPa$ are in figure 6.8. When these results are compared with results obtained using HORUS for a Reynolds number of 150000 and injection times of $\Delta t = 50$ and $Deltat = 100$ in figures 6.9 and 6.11 it is apparent that in both cases deformation of the leading shock is present. In the case of the numerical simulations the deformation of the shock is much greater than that of the experimental results but the trend observed of the shock being significantly more deformed in the case where the injection time is longer is present in both the experimental and numerical data.

Instead if the experimental results are visually compared to simulations, with Reynolds of 6000 in figures 6.10 and 6.12, the overall shape of the distortion is in better agreement. In this case since the turbulent structures present inside the tube are larger in scale and less frequent it appears that the shock is less distorted by these structures.

Although no definitive conclusion can be drawn between the experimental results and the simulations it is evident that some similarities are present. The most notable ones are the distortion of the shock as it travels through the turbulent domain and the weakening of the shock which could be associated with longer injection times. If images of the turbulent structures prior to the shock being generated were available it could be possible to determine if the scale and structure of vorticies present agreed with results from simulations.



(a) Injection Time $t_i = 0.49s$

(b) Injection Time $t_i = 2.53s$

Figure 6.8. Schlieren images of showing the density distribution of experimental results for inlet pressure to initial pressure ratio of 2:1 and injection times of $t_i = 0.49s$ and $t_i = 2.53s$.

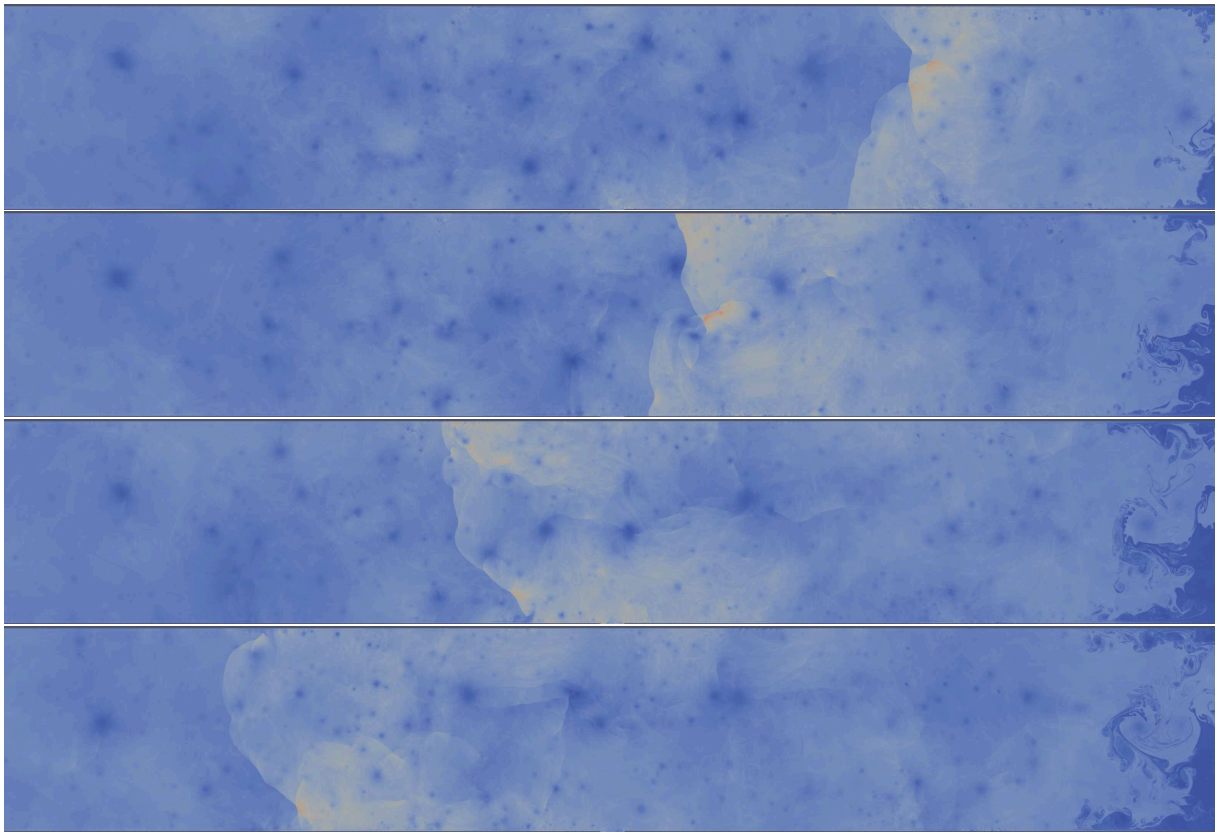


Figure 6.9. Density distribution of the shock traveling through the domain after $\Delta t = 50$ for injection time at a Reynolds number of 150000.

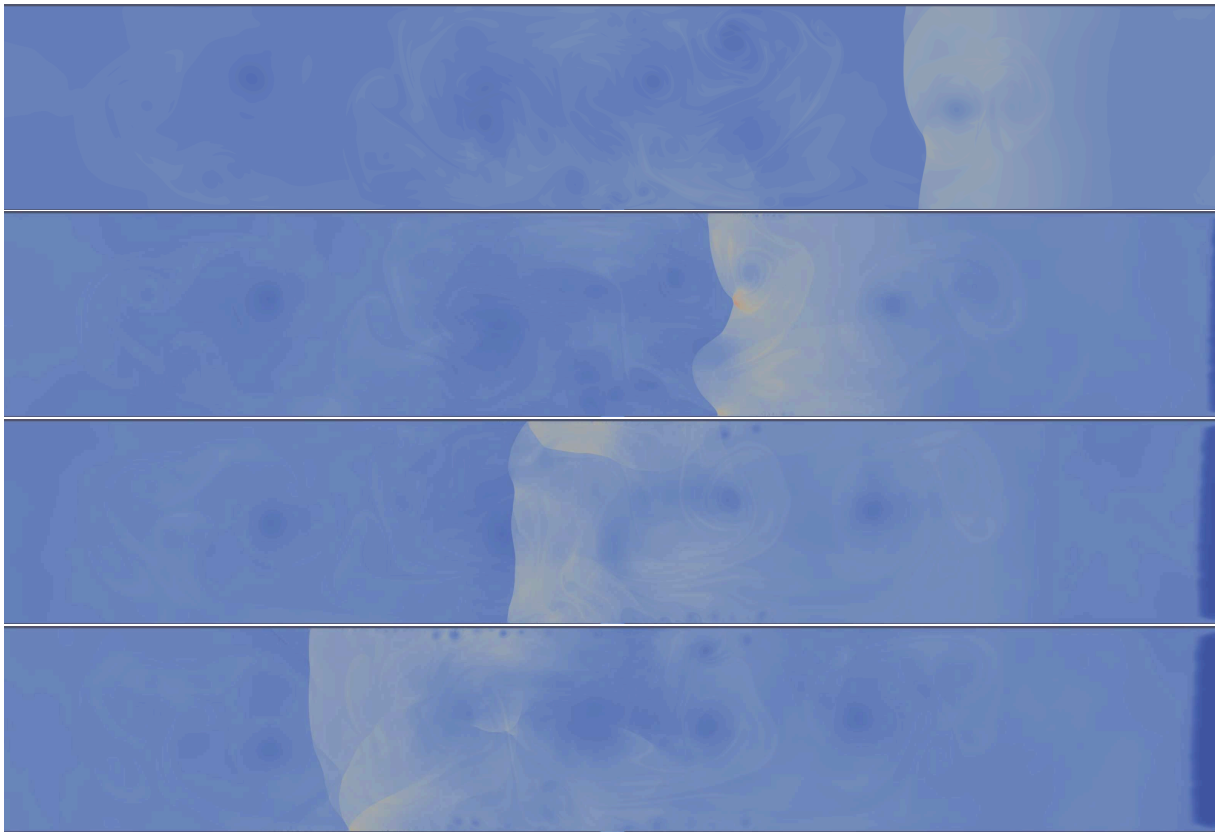


Figure 6.10. Density distribution of the shock traveling through the domain after $\Delta t = 50$ for injection time at a Reynolds number of 6000.

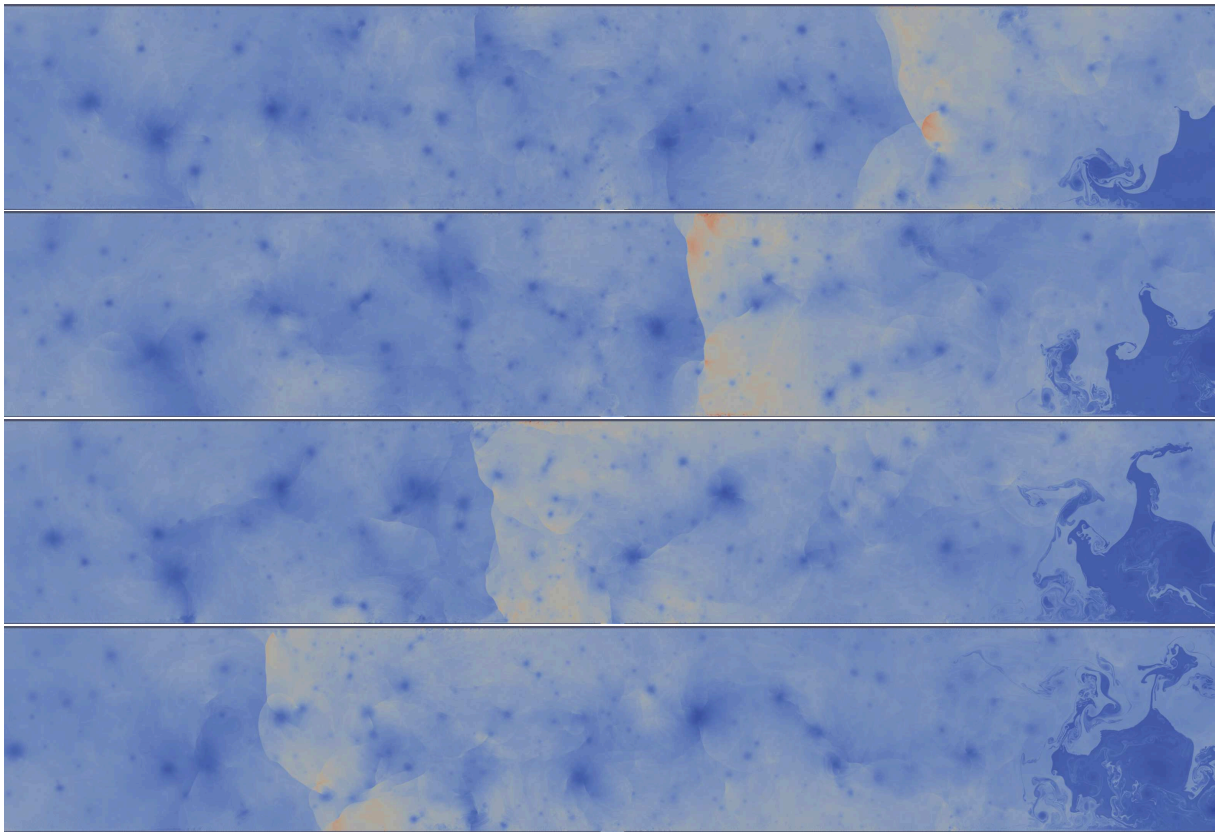


Figure 6.11. Density distribution of the shock traveling through the domain after $\Delta t = 100$ for injection time at a Reynolds number of 150000.

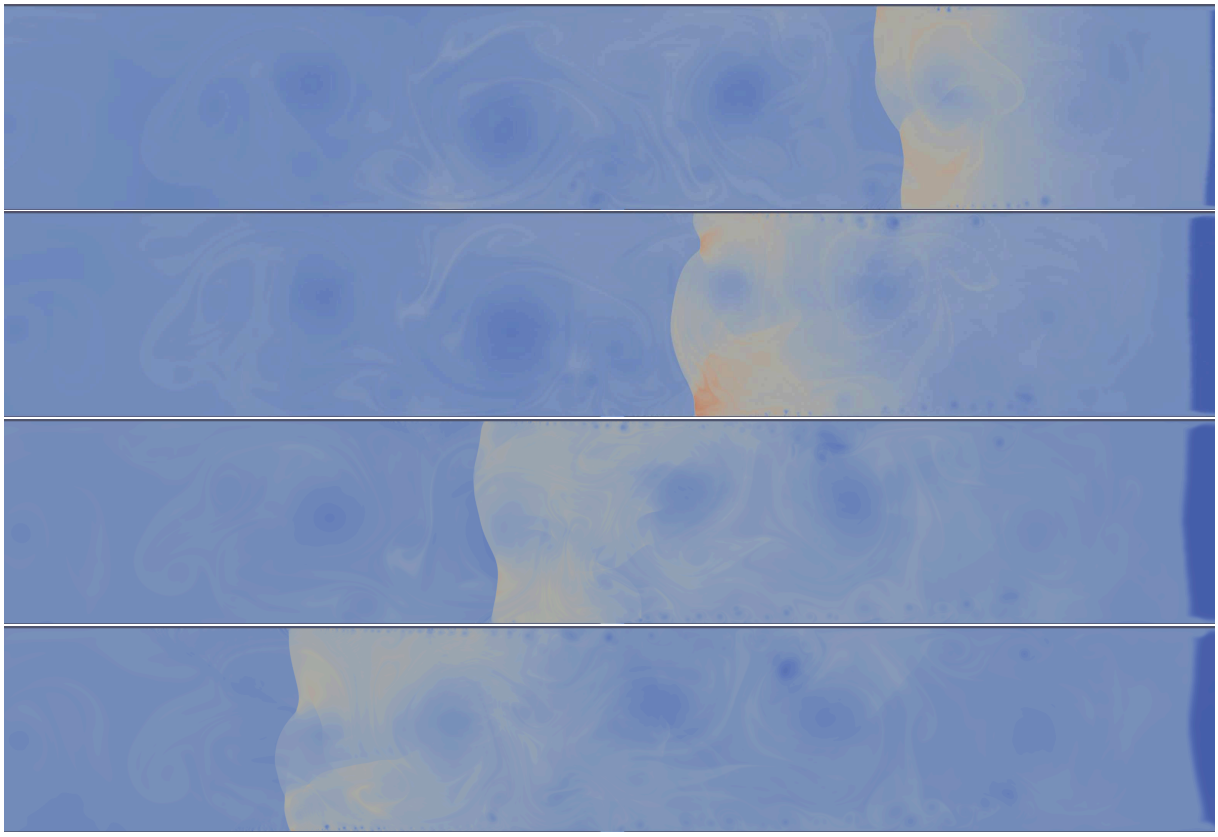


Figure 6.12. Density distribution of the shock traveling through the domain after $\Delta t = 100$ for injection time at a Reynolds number of 6000.

Chapter 7

Conclusion and Future Works

This work investigated the use of DG methods in the context of compressible flow with shocks present. In order to determine the suitability of the DG method in solving problems with discontinuities, various methods were investigated and validated. Since the DG method utilizes a higher order polynomial within the element to obtain higher order accuracy while maintaining a compact stencil, discontinuities within an element need to be handled with care. In the context of this work, the approach used to mitigate the unwanted effects from discontinuities was the addition of an artificial viscosity term to the conservation equation. Additional care needs to be taken to ensure that the term is only added in regions where shocks are present. This is accomplished by using a shock detection mechanism based on the dilation of the velocity.

Preliminary tests were conducted to ensure that the viscosity was not affecting flows that did not contain shocks, unnecessarily damping the solution. After determining that this was indeed the case, investigation into the method's ability to handle simple, one dimensional problems was investigated for varying polynomial degrees. More complex, one dimensional problems involving turbulent like structures, interactions between shocks and rarefaction waves as well as other shocks were also investigated to ensure the solver's robustness and accuracy.

Subsequently, a second spatial dimension was added to determine if significantly more complex flows could be simulated successfully. Although the solver was capable of solving the two dimensional problems and recreated results similar to those obtained by other authors the use of the positive limiter was required for flows involving strong shocks. The simulations performed showed great promise in recreating results obtained on significantly finer meshes using different methods and managed to do so using unstructured meshes, a limitation of several other methods.

It was also found that instead of using mesh refinement techniques, typical of FV methods, an alternative approach could be used to obtain higher accuracy while minimizing computational cost. A polynomial adaptive routine was investigated which consisted of increasing the polynomial degree when shocks are present in order to better resolve the sharp changes in the solution. This method showed a lot of promise in one dimensional cases and gave solutions

similar to uniform $p_s = 8$ using a $p_s = 2, 8$ adaptive routine. This allowed highly accurate results while maintaining the average number of degrees of freedom per element to a low value, significantly reducing computational resources required. Preliminary results for two dimensional cases were promising, but further investigation is required to ensure the robustness of the routine in two dimensions.

After determining the ability of HORUS to resolve benchmark test cases where shocks are present, it was determined that the solver's ability to resolve real world scenarios would be investigated. This was done by comparing laboratory results obtained by sending a shock wave over a turbulent region to numerical results recreating similar conditions. The experimental and numerical results did not give exactly the same density distributions but similarities in the overall structure of the shock wave as it interacted with the turbulent flow field were observed. It was also observed that the shock strength as it traveled through the turbulent area, was consistent when comparing numerical and experimental data.

Building on the findings in this work, further investigation into the adaptive polynomial routine for two dimensional test cases should be considered. The possibility of combining a routine using polynomial and mesh refinement should also be considered to increase the stability of this routine. The solver's ability to handle strong shocks without the need for the positive limiter is also an avenue worth investigating. Finally, a more thorough comparison of the numerical results with the experimental data should be conducted. A more in depth comparison would include the ability to better re-create the final pressure ratios post injection and possibly use different equipment to determine the real turbulence levels within the test tube before the shock is generated.

Bibliography

[1] Numerical analysis: Historical developments in the 20th century. page iv. Elsevier, Amsterdam, 2001.

[2] J.D. Anderson Jr, J. Degroote, G. Degrez, E. Dick, R. Grundmann, and J. Vierendeels. *Computational fluid dynamics*, volume 206. Springer.

[3] H.W. Liepman and A. Roshko. *Elements of Gasdynamics*. Wiley, New York, 1957.

[4] G. Moretti and M. Abbett. A time-dependent computational method for blunt body flows. *AIAA Journal*, 4(12):2136–2141, 1966.

[5] A. Dedner, D. Kröner, C. Rohde, T. Schnitzer, and M. Wesenberg. *Comparison of Finite Volume and Discontinuous Galerkin Methods of Higher Order for Systems of Conservation Laws in Multiple Space Dimensions*. 01 2012.

[6] H.T Huynh. A flux reconstruction approach to high-order schemes including discontinuous galerkin methods. In *18th AIAA Computational Fluid Dynamics Conference*, pages 40–79, 2007.

[7] B. Cockburn. Discontinuous galerkin methods. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 83(11):731–754, 2003.

[8] W.H. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. 1973.

[9] B. Cockburn, G.E. Karniadakis, and C.W. Shu. The development of discontinuous galerkin methods. In *Discontinuous Galerkin Methods*, pages 3–50. Springer, 2000.

[10] D. Radice and L. Rezzolla. Discontinuous galerkin methods for general-relativistic hydrodynamics: formulation and application to spherically symmetric spacetimes. *Phys. Rev. D*, 84:024010, 2011.

[11] E. Tadmor. Shock capturing by the spectral viscosity method. *Computer Methods in Applied Mechanics and Engineering*, 80(1):197–208, 1990.

[12] Y. Jian and J.S. Hesthaven. A study of several artificial viscosity models within the discontinuous galerkin framework. *Communications in Computational Physics*, 27:1309–1343, 06 2020.

[13] D. Moro, N.C. Nguyen, and J. Peraire. Dilation-based shock capturing for high-order methods. *International Journal for Numerical Methods in Fluids*, 82(7):398–416, 2016.

[14] P. Persson and J. Peraire. Sub-cell shock capturing for discontinuous galerkin methods. *AIAA paper*, 2, 01 2006.

[15] A. Burbeau, P. Sagaut, and Ch.H. Bruneau. A problem-independent limiter for high-order runge–kutta discontinuous galerkin methods. *Journal of Computational Physics*, 169(1):111–150, 2001.

[16] Lilia K. Limiters for high-order discontinuous galerkin methods. *Journal of Computational Physics*, 226(1):879–896, 2007.

[17] L. Krivodonova, J. Xin, J.F. Remacle, N. Chevaugeon, and J.E. Flaherty. Shock detection and limiting with discontinuous galerkin methods for hyperbolic conservation laws. *Applied Numerical Mathematics*, 48(3):323–338, 2004. Workshop on Innovative Time Integrators for PDEs.

[18] S. Moe, J. Rossmanith, and D. Seal. A simple and effective high-order shock-capturing limiter for discontinuous galerkin methods. 07 2015.

[19] H. Hoteit, P. Ackerer, R. Mose, J. Erhel, and B. Philippe. New two-dimensional slope limiters for discontinuous galerkin methods on arbitrary meshes. *International Journal for Numerical Methods in Engineering*, 61:2566 – 2593, 12 2004.

[20] M. Sonntag and C.D. Munz. Shock capturing for discontinuous galerkin methods using finite volume subcells. *Finite Volumes for Complex Applications VII*, 78:945–953, 01 2014.

[21] K. Asthana, M. López Morales, and A. Jameson. Non-linear stabilization of high-order flux reconstruction schemes via fourier-spectral filtering. *Journal of Computational Physics*, 303:269–294, 10 2015.

[22] A. Ferrero and F. Larocca. Feedback filtering in discontinuous galerkin methods for euler equations. *Progress in Computational Fluid Dynamics, An International Journal*, 16:14, 01 2016.

[23] J.S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.

[24] D. Gottlieb and J.S. Hesthaven. Spectral methods for hyperbolic problems. *Journal of Computational and Applied Mathematics*, 128(1-2):83–131, 2001.

[25] A. Kanevsky, M.H. Carpenter, and J.S. Hesthaven. Time-consistent filtering in spectral and spectral element methods. *J. Comput. Phys., submitted for publication*, 2004.

[26] J. Glaubitz, Jr. Nogueira, J. Almeida, R. Cantão, and C. Silva. Smooth and compactly supported viscous sub-cell shock capturing for discontinuous galerkin methods. *Journal of Scientific Computing*, 10 2018.

[27] J.H. Choi, J.J. Alonso, and E. van der Weide. A simple and robust shock-capturing approach for discontinuous galerkin discretizations. *Energies*, 12(14), 2019.

[28] J. Markert, G. Gassner, and S. Walch. A sub-element adaptive shock capturing approach for discontinuous galerkin methods. *Communications on Applied Mathematics and Computation*, 08 2021.

[29] B. Cockburn. *Discontinuous Galerkin Methods for Convection-Dominated Problems*, pages 69–224. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[30] T. Haga, M. Furudate, and K. Sawada. *RANS Simulation Using High-Order Spectral Volume Method on Unstructured Tetrahedral Grids*. 2009.

[31] T. Haga, H. Gao, and Z. J. Wang. A high-order unifying discontinuous formulation for the navier-stokes equations on 3d mixed grids. *Mathematical Modelling of Natural Phenomena*, 6(3):28–56, 2011.

[32] Xiaogang D., Yaobing M., Meiliang M., Huayong L., Guohua T., and Hanxin Z. Further studies on geometric conservation law and applications to high-order finite difference schemes with stationary grids. *Journal of Computational Physics*, 239:90–111, 2013.

[33] D. Gottlieb and E. Tadmor. The cfl condition for spectral approximations to hyperbolic initial-boundary value problems. *Mathematics of Computation*, 56(194):565–588, 1991.

[34] L. Krivodonova and R. Qin. An analysis of the spectrum of the discontinuous galerkin method. *Applied Numerical Mathematics*, 64:1–18, 2013.

[35] W.L. Oberkampf and T.G. Trucano. Verification and validation in computational fluid dynamics. *Progress in aerospace sciences*, 38(3):209–272, 2002.

[36] B.C. Vermeire and P.E. Vincent. On the behaviour of fully-discrete flux reconstruction schemes. *Computer Methods in Applied Mechanics and Engineering*, 315:1053–1079, 2017.

[37] C.A. Pereira. Analysis of high-order element types for implicit large eddy simulation. Master's thesis, Concordia University, Montreal, Canada, July 2019. Unpublished.

[38] E.F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.

[39] A. Bhagatwala and S.K. Lele. A modified artificial viscosity approach for compressible turbulence simulations. *J. Comput. Phys.*, 228(14):4965–4969, aug 2009.

[40] Sachin P., Chunlei L., and Antony J. Computation of flows with shocks using the spectral difference method with artificial viscosity, ii: Modified formulation with local mesh refinement. *Computers & Fluids*, 98:122–133, 2014. 12th USNCCM mini-symposium of High-Order Methods for Computational Fluid Dynamics - A special issue dedicated to the 80th birthday of Professor Antony Jameson.

[41] J.H. Choi, J.J. Alonso, and E.V.D. Weide. *Simple shock detector for discontinuous Galerkin method*.

[42] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54(1):115–173, 1984.

[43] E.F. Toro and M.E. Vázquez-Cendón. Flux splitting schemes for the euler equations. *Computers & Fluids*, 70:1–12, 2012.

[44] U.S. Vevek, B. Zang, and T.H. New. On alternative setups of the double mach reflection problem. *Journal of Scientific Computing*, pages 1–13, aug 2018.

[45] S. Zhao. *Explicit and implicit large eddy simulation of turbulent combustion with multi-scale forcing*. Theses, Université d'Orléans, May 2016.

[46] P. Persson. *Shock Capturing for High-Order Discontinuous Galerkin Simulation of Transient Flow Problems*.

[47] Friedemann K. On the proper setup of the double mach reflection as a test case for the resolution of gas dynamics codes. *Computers & Fluids*, 132:72–75, 2016.

[48] G. Ben-Dor. Analytical solution of double-mach reflection. *AIAA Journal*, 18(9):1036–1043, 1980.

[49] N. Nikiforakis. The euler equations with a single-step arrhenius reaction. 2013.

[50] J.H.W. Lee and V.H. Chu. *Turbulent Jets and Plumes: A Lagrangian Approach*, chapter Turbulent Jets, pages 21–54. Springer, 2003.

[51] B.J. Marjaba. Impact of turbulence on detonation dynamics. Master's thesis, Concordia University, Montreal, Canada, April 2021.